

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Comunicación  
Inalámbrica para Aplicaciones de Robótica de Enjambre -  
Fase II**

Trabajo de graduación presentado por Hee Chan Moisés Kim Cabrera  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Comunicación  
Inalámbrica para Aplicaciones de Robótica de Enjambre -  
Fase II**

Trabajo de graduación presentado por Hee Chan Moisés Kim Cabrera  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022



Vo.Bo.:



(f) \_\_\_\_\_  
MAEB. Pablo Mazariegos

Tribunal Examinador:



(f) \_\_\_\_\_  
MAEB. Pablo Mazariegos



(f) \_\_\_\_\_  
MSc. Luis José Pinillos



(f) \_\_\_\_\_  
Dr. Luis Alberto Rivera Estrada

Fecha de aprobación: Guatemala, 5 de enero de 2022.



El presente trabajo surge con la curiosidad de llevar la robótica de enjambre a un entorno real y no únicamente simulaciones. Se ha escuchado de las muchas aplicaciones que puede tener la robótica de enjambre en varios ámbitos desde la mecánica automotriz hasta búsqueda y rescate, sin embargo, en la actualidad no hay una tecnología concreta con la que se puedan cumplir estas aplicaciones. Es por ello, que construir el sistema de comunicación para un enjambre acerca la posibilidad de ver las aplicaciones de esta tecnología no solamente en simulaciones sino que en robots reales ayudando a los futuros investigadores a cumplir tareas complejas.

Quisiera tomar este espacio para darle las gracias a todos aquellos que me han apoyado y han tomado el tiempo y el esfuerzo para que este trabajo se llevará acabo.

Primero, a Dios por darme la sabiduría y la capacidad para poder completar este trabajo.

A mis padres, por exigirme a dar lo mejor en cada etapa de mi vida y enseñarme a mantener mis principios y valores en todo lo que haga.

A mi asesor, MBAE Ing. Pablo Mazariegos por brindarme su tiempo, sus consejos y apoyo en el desarrollo de este trabajo. También al Dr. Luis Rivera por el apoyo que me ha dado y compartir conmigo sus conocimientos.

A mis amigos, con los que durante todo el proceso de desarrollo de este trabajo fuimos apoyándonos y animándonos a completar el trabajo.



<b>Prefacio</b>	<b>v</b>
<b>Lista de figuras</b>	<b>xii</b>
<b>Lista de cuadros</b>	<b>xiii</b>
<b>Resumen</b>	<b>xv</b>
<b>Abstract</b>	<b>xvii</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
2.1. E-pucks como una herramienta educativa para robótica de enjambre . . . . .	4
2.2. Fases previas en la UVG . . . . .	5
<b>3. Justificación</b>	<b>7</b>
<b>4. Objetivos</b>	<b>9</b>
4.1. Objetivo general . . . . .	9
4.2. Objetivos específicos . . . . .	9
<b>5. Alcance</b>	<b>11</b>
<b>6. Marco teórico</b>	<b>13</b>
6.1. Robótica de enjambre . . . . .	13
6.2. Modelo de red de siete capas . . . . .	13
6.3. Arquitectura centralizada . . . . .	15
6.4. Arquitectura descentralizada . . . . .	15
6.5. ZigBee . . . . .	15
6.6. Fallas en la red ZigBee . . . . .	19
6.7. Protocolo 802.15.4 . . . . .	20
6.8. <i>Broadcast</i> . . . . .	21
6.9. Estudio de compensación . . . . .	21

6.10. Métodos de sincronización en redes inalámbricas de sensores . . . . .	21
6.10.1. <i>Reference broadcast synchronization</i> . . . . .	22
6.10.2. <i>Time sync protocol for sensor networks</i> . . . . .	23
6.10.3. <i>Flooded time synchronization protocol</i> . . . . .	23
6.10.4. <i>Time Synchronize mesh protocol</i> . . . . .	23
6.11. Algoritmos de elección de coordinador . . . . .	24
6.11.1. Algoritmo bravucón . . . . .	24
6.11.2. Algoritmo de anillo . . . . .	25
6.11.3. Algoritmo adicional . . . . .	25
6.12. Rendimiento en una red . . . . .	26
6.12.1. Rendimiento en redes descentralizadas . . . . .	26
6.12.2. Rendimiento en la red Zigbee . . . . .	27
<b>7. Elección de topología, tecnología y hardware</b>	<b>31</b>
7.1. Criterios de elección . . . . .	31
7.2. Topología . . . . .	32
7.2.1. Aspectos cualitativos de una topología . . . . .	32
7.2.2. Topologías evaluadas . . . . .	33
7.2.3. Elección de la topología para robótica de enjambre . . . . .	34
7.3. Tecnología . . . . .	35
7.3.1. Tecnologías compatibles con la topología seleccionada . . . . .	35
7.3.2. Elección de la tecnología para la topología seleccionada . . . . .	35
7.4. Transceptor . . . . .	36
7.4.1. Propuestas compatibles con la tecnología seleccionada . . . . .	36
7.4.2. Elección del transceptor compatible con la tecnología seleccionada . . . . .	38
<b>8. Diseño de la red</b>	<b>39</b>
8.1. Supuestos . . . . .	39
8.2. Funciones básicas . . . . .	39
8.3. Servicios . . . . .	42
8.3.1. Servicio de asociación . . . . .	42
8.3.2. Servicio de sincronización de mensajes . . . . .	46
8.3.3. Algoritmo de nodos conectados . . . . .	49
8.3.4. Servicio de elección de coordinador . . . . .	51
<b>9. Medición de la red</b>	<b>55</b>
9.1. Metodología para la tasa de rendimiento efectiva . . . . .	56
9.2. Metodología para el consumo energético . . . . .	58
9.3. Metodología para el alcance y retardo en paquetes . . . . .	58
<b>10. Resultados</b>	<b>61</b>
10.1. Tasa de rendimiento efectiva . . . . .	61
10.2. Alcance y retardo de paquetes . . . . .	62
10.3. Consumo energético . . . . .	64
10.4. Efectividad de los servicios . . . . .	66
<b>11. Conclusiones</b>	<b>69</b>
<b>12. Recomendaciones</b>	<b>71</b>

<b>13. Bibliografía</b>	<b>73</b>
<b>14. Anexos</b>	<b>77</b>
14.1. Librería para el transceptor MRF24J40MA . . . . .	77



---

## Lista de figuras

---

1.	Configuraciones de robots en un enjambre [2]. . . . .	3
2.	E-puck comercial [3]. . . . .	4
3.	Red de comunicación wifi propuesta en la fase previa [4]. . . . .	5
4.	Capas de 802.15.4 y ZigBee [9]. . . . .	16
5.	Estructura de una supertrama [9]. . . . .	17
6.	Sistema de sincronización con RBS. . . . .	22
7.	Operación del algoritmo de anillo [18] . . . . .	25
8.	Un paquete de ida y vuelta . . . . .	27
9.	Circuito para el consumo energético [23]. . . . .	29
10.	Estudio de compensación de la topología. . . . .	34
11.	Estudio de compensación de las tecnologías. . . . .	36
12.	Módulo Xbee [38]. . . . .	37
13.	Módulo MRF24J40MA [11]. . . . .	37
14.	<i>Shield</i> desarrollado para Arduino Nano. . . . .	38
15.	Diagrama de la clase MRF24J40MA. . . . .	41
16.	Secuencia de comunicación para el servicio de asociación. . . . .	42
17.	Algoritmo del coordinador. . . . .	44
18.	Algoritmo del nodo. . . . .	45
19.	Matriz para una red con espacios de tiempo y 5 canales disponibles [15]. . . . .	46
20.	Secuencia de comunicación para el servicio de sincronización. . . . .	47
21.	Algoritmos del servicio de sincronización. . . . .	48
22.	Secuencia de comunicación para el algoritmo de nodos conectados para una red con un nodo . . . . .	49
23.	Algoritmos de nodos conectados. . . . .	50
24.	Algoritmo del nodo. . . . .	52
25.	Algoritmo del candidato. . . . .	53
26.	Algoritmo del emisor para la tasa de rendimiento efectiva. . . . .	56
27.	Algoritmo del receptor para la tasa de rendimiento efectiva. . . . .	57
28.	Circuito para medir la corriente consumida. . . . .	58
29.	Proceso de transmisión de paquete [23]. . . . .	59

30.	Histograma para la tasa de transferencia. . . . .	62
31.	Diagrama de dispersión con una distancia de 2 metros. . . . .	63
32.	Diagrama de dispersión con una distancia de 6 metros. . . . .	63
33.	Diagrama de dispersión con una distancia de 10 metros. . . . .	64
34.	Medición del voltaje para la transmisión con el transceptor MRF24J40MA. . .	65
35.	Medición del voltaje para la recepción con el transceptor MRF24J40MA. . . .	65

---

## Lista de cuadros

---

1.	Comparación de tecnologías de comunicación en fases previas [4]. . . . .	6
2.	Tecnologías para topologías descentralizadas. . . . .	35
3.	Cotización para el módulo Xbee. . . . .	37
4.	Cotización para el módulo MRF34J40MA. . . . .	37
5.	Comparación de distancias en un espacio cerrado. . . . .	64
6.	Comparación de éxitos para el servicio de asociación. . . . .	66
7.	Comparación de éxitos para el servicio de sincronización. . . . .	66
8.	Comparación de éxitos para el algoritmo que revisa los nodos conectados. . .	67
9.	Resultado del servicio para seleccionar nuevo coordinador. . . . .	67



La transición de simulación a realidad para cualquier ámbito presenta un reto y es el caso también para la robótica de enjambre. Por lo tanto, en esta investigación se buscó acortar la brecha entre la simulación y la realidad enfocándose en la manera que los robots se comunican. La transición a la realidad de la forma en que se comunican los robots trae consigo problemas relacionados a las redes de comunicación. El enjambre, que en simulación no presenta problemas de colisión de datos, en sistemas físicos sí los presenta. La velocidad de comunicación entre robots está limitada, y el fallo de un robot puede significar el fallo del enjambre completo.

Para resolver estos problemas se compararon, se evaluaron y se seleccionó una topología y una tecnología de comunicación inalámbrica. Se seleccionaron para construir una red capaz de cumplir con los criterios de robótica de enjambre. Además, se buscó que la red pueda mitigar o eliminar los problemas inherentes a las redes que pueden afectar la robótica de enjambre.

Adicionalmente, se midió la red para determinar sus parámetros como la velocidad de transferencia y la robustez para brindar a los futuros investigadores una red que puedan implementar en aplicaciones de robótica de enjambre.



The transition from simulation to reality for any domain is challenging and this is the case for swarm robotics as well. Therefore, this research attempts to shorten the gap between simulation and reality by focusing on the way robots communicate. The transition to reality in the way robots communicate brings with it problems related to communication networks. The swarm that in simulation did not present data collision problems now presents them, the communication speed is now limited and most importantly the failure of a single robot can lead to the failure of the entire swarm.

To solve these problems, a topology was compared, evaluated and selected, a wireless communication technology was selected to build a network capable of meeting the criteria of swarm robotics and mitigate or eliminate the problems inherent in networks that can affect the swarm.

Additionally, the network was measured to determine its parameters such as transfer speed and robustness to provide future researchers with a network that they can implement in swarm robotics applications.



En este trabajo se presenta una alternativa a la topología seleccionada en un trabajo de graduación anterior, que era una topología de estrella. Una de las motivaciones para presentar una alternativa es que el comportamiento colectivo del enjambre se encontraba en peligro por cualquier falla del *router*. Con la alternativa propuesta se pretende tener una mayor fiabilidad de los algoritmos de inteligencia de enjambre al brindar una red más robusta. Para alcanzar esta meta se establecieron los objetivos enfocados a seleccionar y diseñar la red alterna.

Como metodología, primero se tiene comparar las características que describen una topología para todas las topologías propuestas y utilizar las características de robótica de enjambre como criterio para seleccionar la topología más adecuada. Se realiza la comparación para las distintas tecnologías de comunicación con las mismas características para tener la más adecuada. Luego, como se busca tener una aplicación real, se seleccionaron y se compraron los transceptores necesarios para construir una red física. Con la topología, tecnología y transceptores definidos se prosiguió a construir la red. Se desarrollaron las funciones necesarias para la comunicación entre el microcontrolador y el transceptor. También, se desarrollaron los servicios necesarios para resolver los problemas inherentes a las redes de comunicación. Finalmente, se realizaron experimentos para medir el rendimiento de la red y brindar una estructura completa del sistema de comunicación para futuras investigaciones.



La robótica de enjambre se volvió una rama de la robótica que llamó el interés de varias instituciones y organizaciones para desarrollar sistemas eficientes y lograr objetivos que se pueden alcanzar con inteligencia colectiva. Es por ello, grupos como el Ecole Polytechnique Fédérale de Lausanne[1] y el departamento de ingeniería electrónica y mecatrónica de la UVG, entre otros, han realizado investigaciones enfocadas en la robótica para trasladar simulaciones de robótica de enjambre al mundo real. Por una parte, el Grupo Ecole ha delimitado características y parámetros que debe cumplir una plataforma física mientras que la UVG, por su lado, ha propuesto una red WiFi para implementarlo con robots físicos.

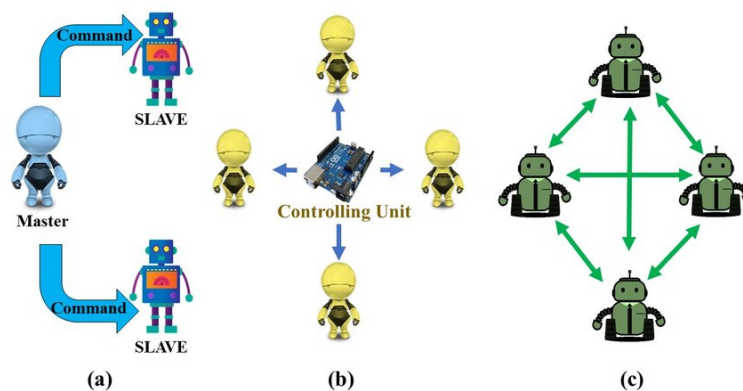


Figura 1: Configuraciones de robots en un enjambre [2].

## 2.1. E-pucks como una herramienta educativa para robótica de enjambre

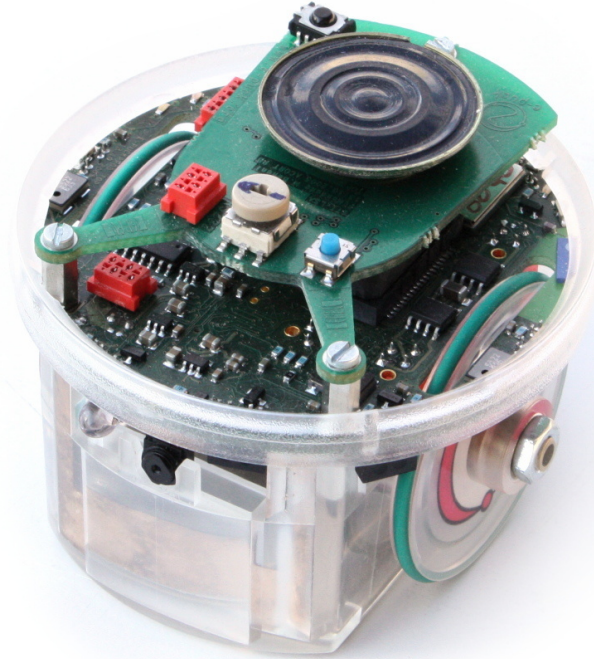


Figura 2: E-puck comercial [3].

En el paper de Group Ecole Polytechnique Fédérale de Lausanne de Suiza [1] se enfatizó la transición de las simulaciones a la realidad ya que los retos en la realidad son distintos. Se definieron propiedades de la robótica de enjambre que son claves para tener un verdadero sistema de inteligencia de enjambre. Las propiedades son:

- Auto-organización
- Escalabilidad
- Robustez

A partir de estas propiedades buscaron una solución de comunicación que les permitiera mantenerlas. Definieron una comunicación local entre robots donde no era necesario un servidor al cual todos los robots debían comunicarse. Los robots utilizados, E-pucks, tenían un rango de comunicación de 15 cm que les permitía comunicarse con otros agentes del enjambre.

Adicionalmente, expandieron el tema de auto-organización y se muestra el rol que cumple el sistema de comunicación.

Principios de la auto-organización:

- Retroalimentación positiva: amplificación, para reclutamiento y refuerzo.
- Retroalimentación negativa: para balancear los mecanismos como saturación o competencia.
- Aleatoriedad: lo impredecible es importante para la exploración.
- Múltiples interacciones: tiene que existir un número mínimo de agentes capaces de reaccionar ante la presencia y movimiento de otros.

## 2.2. Fases previas en la UVG

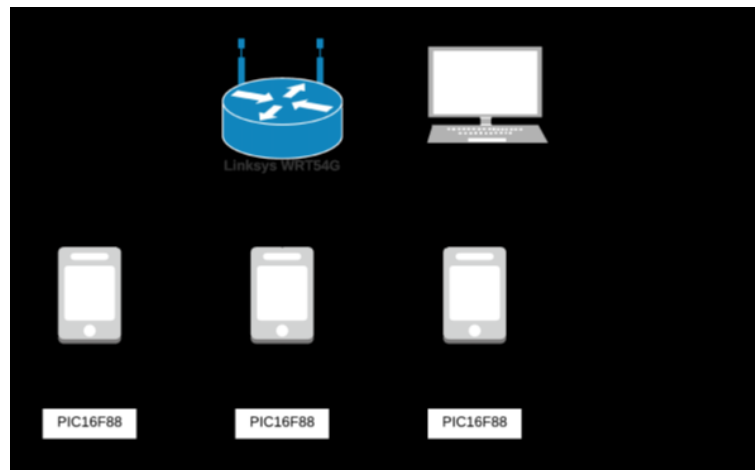


Figura 3: Red de comunicación wifi propuesta en la fase previa [4].

En la fase previa desarrollada en el 2019, “Diseñar e implementar una red de comunicación inalámbrica para la experimentación en robótica de enjambre” [4] se utilizó una topología de estrella con la tecnología WiFi para la comunicación entre robots. El dispositivo centralizador seleccionado fue el Router Linksys WRT54GL y los agentes utilizaban ESP8266 como modulo de comunicación y PIC16F88 como MCU para una red con 254 clientes. Los parámetros obtenidos con la red son:

- 11.7 Mbps para la velocidad de transmisión máxima.
- 7.70 Kbps para la comunicación entre el ESP8266 y PIC16F88 en una comunicación de 9600 baudios.

El Cuadro 1 muestra la comparación de tecnologías de comunicación en fases previas.

<b>Características</b>	<b>Bluetooth</b>	<b>WiFi</b>	<b>LTE</b>
Cantidad de nodos por red	7	255 con direcciones clase C	>200 con celdas de 5MHz
Velocidad de transmisión	1-24 Mbps	11-100 Mbps	20-100 Mbps
Costo de Implementación	Bajo	Bajo	Alto
Ancho de banda	2.4 GHz	2.4 GHz o 5 GHz	Bandas licenciadas
Uso	Mouse, teclado auriculares	Redes de computadora	Telefonía móvil

Cuadro 1: Comparación de tecnologías de comunicación en fases previas [4].

El sistema de comunicación desarrollado anteriormente presentó la limitante de la robustez que es una característica de robótica de enjambre, ya que el sistema requiere de un *master* para coordinar la comunicación de todos los clientes en la red. Por lo que, si llegara a funcionar incorrectamente el punto de acceso, toda la red se vendría a bajo. Por esta razón, se pretende presentar redes alternativas que sean descentralizadas y poder cumplir con una comunicación que satisfaga todas las características de robótica de enjambre.

Maximizar ganancias o minimizar pérdidas ha sido siempre el énfasis de resolver los problemas ingenieriles y para diversas ramas de la ciencia los problemas de optimización han ido aumentando su nivel de complejidad con el avance de la tecnología. Por este motivo, surgió la robótica de enjambre que es la coordinación de sistemas de agentes que interactúan entre si y con el entorno acorde a una inteligencia de enjambre. Los distintos algoritmos que se están desarrollando en la inteligencia de enjambre como el algoritmo de optimización por enjambre de partículas (PSO, por sus siglas en inglés), algoritmo de optimización por colonia de hormigas (ACO, por sus siglas en inglés), entre otros, han servido como herramientas para resolver problemas estocásticos a un nivel de simulación. Sin embargo, con la visión de implementar esta tecnología de manera física como respuesta a resolver tareas complejas con un diseño económico se enfrenta a un reto que es intrínseco de una implementación física, la comunicación entre robots. A diferencia de una simulación donde se itera una función o un algoritmo que es equiparable a la comunicación entre robots, en la realidad se necesita traducir las iteraciones a interacciones entre robots y es esta traducción al mundo físico que requiere de un sistema de comunicación entre los robots para llevar a cabo los algoritmos a fin de encontrar la solución óptima.

Por este motivo, el enfoque principal del proyecto es proponer sistemas de comunicación descentralizada que cumplan con las características de la robótica de enjambre: auto-organización, escalabilidad y robustez [1]. Dadas estas tres características de la robótica de enjambre se proponen sistemas de comunicación alternativos al presentado en la fase I, red WiFi con topología de estrella, ya que este sistema no cumple con la robustez. Una red WiFi con topología de estrella requiere de un router para que todos los hosts que serían los robots pudieran comunicarse, por ende, si llega a fallar el router se cae toda la red y los robots ya no pueden comunicarse. Entonces la propuesta de una red descentralizada busca solucionar este problema al brindar una comunicación entre pares en donde si falla un robot, simplemente dejan de comunicarse con él.

Finalmente, al proponer sistemas de comunicación descentralizada se tiene la intención de construir distintas opciones que implementan distintos protocolos en cada capa del modelo

de referencia OSI [5] con el fin de utilizar la red que mejor se adecúe a un algoritmo de inteligencia de enjambre.

### 4.1. Objetivo general

Diseñar e implementar sistemas de comunicación inalámbrica alternativos a los propuestos en la fase anterior basados en una topología descentralizada para aplicaciones de robótica de enjambre.

### 4.2. Objetivos específicos

- Comparar la topología de estrella propuesta en la fase previa con topologías descentralizadas y seleccionar las más adecuadas para las aplicaciones de robótica de enjambre en la UVG.
- Comparar y seleccionar el protocolo de comunicación que mejor se adecúe a la topología seleccionada.
- Diseñar e implementar el sistema de comunicación en la mesa de pruebas de UVG.
- Determinar el rendimiento del sistema de comunicación propuesto en sistemas de múltiples robots.



El alcance de este trabajo abarcó investigar las distintas topologías y tecnologías para poder compararlas y definir las más adecuadas para aplicaciones de robótica de enjambre. La deliberación de la tecnología y topología se realizó únicamente por un proceso de investigación y comparación para implementar físicamente la opción más adecuada. Además, este trabajo se centró en el diseño e implementación de la red basada en la topología y tecnología seleccionada para tener como resultado una estructura que se puede utilizar para un sistema de comunicación.

El diseño de la red abarcó crear una librería completa para manejar el transceptor seleccionado y tener todos los servicios necesarios para que los problemas que se presenten en la red relacionados a las características de robótica de enjambre: robustez, autoorganización y escalabilidad; se resuelvan automáticamente en la red sin necesidad de resolver el problema reiniciando manualmente o reprogramando los microcontroladores.



## 6.1. Robótica de enjambre

La robótica de enjambre es el estudio de cómo diseñar grupos de robots que operan sin depender de una estructura externa o alguna forma de control centralizado. El comportamiento colectivo de los robots resultan de una interacción local entre ellos y con el entorno. El diseño de robots de enjambre es guiado por los principios de la inteligencia de enjambre. Estos principios promueven la realización de sistemas que son tolerantes a las fallas, escalables y flexibles. La robótica de enjambre resulta ser útil cuando diferentes actividades deben realizarse con recurrencia y cuando técnicamente no es viable establecer la infraestructura requerida para controlar los robots de una manera centralizada [6].

La robótica de enjambre es tolerante a fallas debido a que se desarrollan sistemas en los que la pérdida de un individuo no significa la falla del enjambre completo. Además, es escalable por los sensores locales y la comunicación ya que los robots continuarán comunicándose con aproximadamente la misma cantidad de compañeros a pesar de que un robot sea removido. Finalmente, la auto-organización recae en la distribución y la natural organización propia de la robótica de enjambre: en enjambre, los robots se posicionan a ellos mismos dinámicamente en las distintas tareas para cumplir con los requisitos del ambiente y las condiciones de operación. Por eso los robots operan de forma básica utilizando los sensores y comunicándose entre ellos para no depender un infraestructura preexistente o alguna información global.

## 6.2. Modelo de red de siete capas

Es importante establecer primeramente los conceptos de protocolo y capas. Las capas son divisiones conceptuales que agrupan funciones similares y los protocolos son agrupaciones de reglas que se han acordado. El modelo Interconexión de Sistemas Abiertos (OSI) es un marco conceptual que describe las funciones de un sistema de red o de telecomunicaciones.

La descripción de este modelo están descritos de la capa superior a la inferior [5].

## **7) Aplicación**

Es la capa que la mayoría de usuarios ven ya que es la “más cercana al usuario final”. Las funciones de esta capa son la identificación de compañero de comunicación, determinación de recursos disponibles y sincronización de comunicación. Para transmitir datos la capa identifica la identidad y disponibilidad del destinatario; luego, se determinan los recursos disponibles para ver si son suficientes para la comunicación.

## **6) Presentación**

Esta capa representa la preparación o traducción del formato de la aplicación al formato de red o viceversa. Un ejemplo es el cifrado y decifrado de datos para una transmisión segura.

## **5) Sesión**

Cuando dos dispositivos se comunican se requiere una sesión. Las funciones de esta capa implican la configuración, coordinación y terminación entre las aplicaciones en cada extremo de sesión.

## **4) Transporte**

Se ocupa de la coordinación de la transferencia de datos entre los sistemas finales y hosts. Cuántos datos enviar, a qué velocidad, a donde va, etc.

## **3) Red**

Esta capa es la encargada del reenvío de paquetes, incluido el enrutamiento. Se define la dirección de la red. En el protocolo de internet se definen direcciones de forma que la selección de la ruta se puede determinar sistemáticamente mediante la comparación de la dirección de la red fuente con la dirección de la red de destino.

## **2) Enlace de datos**

Proporciona transferencia de datos de nodo a nodo y también maneja la corrección de errores de la capa física. Existen dos subcapas: la capa de control de acceso al medio (MAC) y la capa de control de enlace lógico (LLC).

## 1) Física

Es la representación eléctrica y física del sistema. Incluye desde el tipo de cable, el enlace de radio frecuencia, así como la distribución de los pines, voltajes y otros requisitos físicos.

### 6.3. Arquitectura centralizada

Es una de las topologías más populares para un sistema de redes. Se requiere como mínimo un nodo que este conectado directamente con un servidor central. Es un sistema con clientes y servidores donde todas las funciones e información está centralizada. En la infraestructura de la red el servidor es el encargado de consolidar la información recibida de los clientes al tener un hub en común para todos. Una vez conectados los clientes todos los recursos están a disposición de ellos a través del servidor. La arquitectura centralizada presenta la facilidad de retransmitir la información al sistema por tener el diseño de una sola base de datos. Una base de datos vuelve al sistema centralizado más manejable y confiable en la información que transmite [7].

### 6.4. Arquitectura descentralizada

En una red de pares o *peer to peer* donde todos los nodos se comunican proporcionalmente por igual. Cada nodo en este tipo de red toma sus propias decisiones y el comportamiento general de la arquitectura está estimado por la acumulación de decisiones realizadas por cada nodo. Este tipo de arquitectura es popular por la robustez y extensibilidad que le permite a cualquier cantidad de nodos crear archivos en el sistema a pesar que se trabaje únicamente en una red local ya que no hay cambios significativos si se trabaja en una red offline u online [7].

### 6.5. ZigBee

Zigbee es un estándar que define un protocolo de comunicación para baja velocidad y una red de corto alcance, pero es de bajo consumo de potencia. Los dispositivos basados en ZigBee operan en las frecuencias de 868 MHz, 915 MHz y 2.4 GHz. Puede llegar a transmitir hasta 250 Kbps. Esta tecnología esta enfocada hacia aplicaciones que requieran costo bajo, baja transmisión de datos y el mayor tiempo de vida posible usando baterías [8].

Zigbee desarrolla un protocolo basado en el estándar IEEE 802.15.4 para sus primeras dos capas, capa física y subcapa de acceso al medio. En la Figura 4 se presenta las capas del protocolo ZigBee. Este modelo presenta siete capas, pero ZigBee utiliza solo cuatro capas con el objetivo de simplificar la arquitectura. Las primeras dos capas que son la física y la de acceso al medio, luego, se definen las capas de red y de aplicación. Cada una de las capas se conecta con las capas adyacentes por medio de un SAP (Service Access Point) [9].

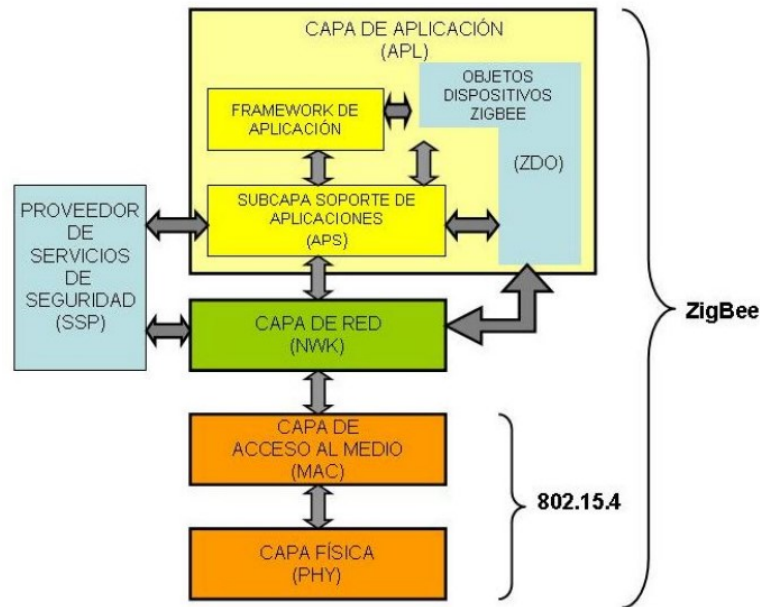


Figura 4: Capas de 802.15.4 y ZigBee [9].

## La capa física

Corresponde a IEEE 802.15.4 que define aspectos de potencia del transmisor y sensibilidad del receptor. En lo que respecta los canales representado por frecuencias, existe un límite de 27 canales con 1 canal en la banda de 868 MHz, 10 en la banda de 915 MHz y 16 en la banda de 2.4 GHz. Antes de transmitir en un canal, el dispositivo debe medir el nivel de energía en ese canal. Se admite una diferencia de 10dB entre la sensibilidad del receptor y el nivel mínimo de energía detectable. El rango de energía que exige el protocolo es de 40dB. La capa provee el servicio de detección de energía en un canal dado y lo envía a la MAC por medio de un entero. Otro mecanismo para verificar si el canal está libre es el CS (Carrier Sense) que consiste en demodular la señal recibida y determinar si es compatible con el estándar, el canal se considera ocupado si hay una señal compatible [9].

El servicio entre capas usan el concepto de “primitivas” para describir los servicios que una capa le provee a la capa superior. La capa superior usa un servicio de punto de acceso para requerir servicios y las etapas de la comunicación son: pedido, confirmación, respuesta e indicación. Se utiliza la primitiva *PD-Data.indication*

## La capa MAC

Esta capa provee una interfase entra la capa física y la capa sobre ella que en el caso de ZigBee es la de red. Esta capa cuenta con dos partes: una encargada de manejar MLME encargada de manejar servicios y una unidad de datos. La MLME interactúa con las capas vecinas por medio de puntos de acceso. La MAC tiene su propia base de datos donde están todos las constantes y atributos definidos por el estándar IEEE 802.15.4. Debido a que no existe un mecanismo de diálogo que soporte RTS/CTS (request to send/clear to send) no

se puede implementar una solución a nivel de software para asegurar la comunicación entre dos dispositivos [9].

Cuando un dispositivo desea transmitir, previamente verifica que el canal no esté en uso por otro dispositivo. Si está libre comienza a transmitir. Hay transmisiones que se hacen sin verificación previa. Estas son: transmisión de baliza, transmisión durante el período libre de contienda (CFP) y transmisión después de haber dado un ACK a un pedido de datos. Un algoritmo de transmisión utilizado *Carrier sense multiple access with collision avoidance* (CSMA/CA, por sus siglas en inglés) que emplea el concepto de baliza y supertrama. El uso de baliza en la red permite disponer de ranuras de tiempo garantizadas (GTS). Para eso se crean tramas especiales MAC llamadas de baliza y, entonces, es posible usar una estructura llamada supertrama.

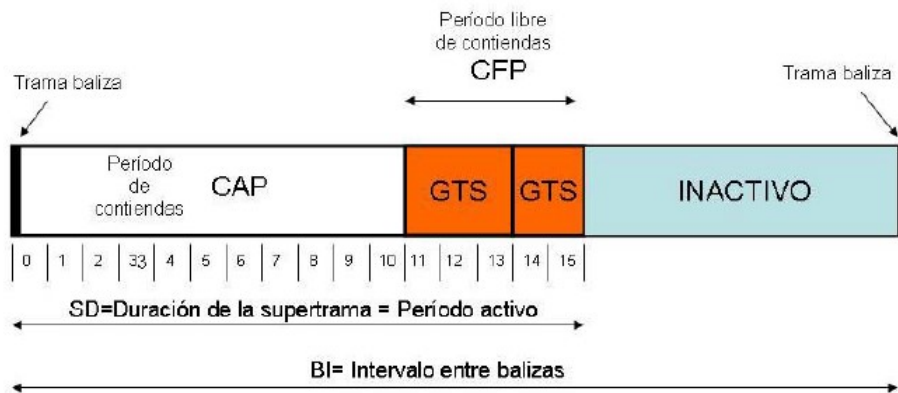


Figura 5: Estructura de una supertrama [9].

La supertrama tiene tres periodos y específicamente en el período de acceso en contienda (CAP) debe de usarse CSMA-CA para acceder a un canal que esté disponible para todos los dispositivos. En el uso de CSMA-CA el tiempo activo se divide en 16 ranuras iguales, entonces el tiempo de back off debe ser alineado para que caiga en el CAP. Cuando no se emplean supertramas, no se necesita sincronizar el back off.

La capa MAC es responsable de:

- Generar balizas si es coordinador.
- Usa CSMA-CA como método de compartir el canal.
- Provee manejo, sincronización y GTS cuando se usan balizas.
- Provee un enlace seguro entre las MACs.
- Provee servicios de asociación desasociación.
- Provee un mecanismo de seguridad cuyo nivel está determinado por lo solicitado desde las capas superiores.

## La capa de red

Esta capa provee funciones para el armado y manejo de redes y una interfaz simple para relacionarla con las aplicaciones del usuario. Provee dos tipos de servicio: de datos a través de la NLDE y de control por medio de la NLME. Cada uno de estos se comunica con la capa de aplicación y la capa MAC por medio de los respectivos puntos de acceso. La capa de red del coordinador asigna direcciones de 16 bits a cada miembro de la red de área personal (PAN). Cada trama de red lleva un parámetro llamado radio que indica la cantidad de saltos máximos que este puede llegar a realizar. Existen tres tipos de comunicación de mensajes:

- Broadcast: el destino es todo dispositivo que lo pueda recibir.
- Multicast: se envía a un grupo en específico.
- Unicast: contiene la dirección de un único dispositivo.

El estándar define dos tipos de dispositivos: Dispositivos de funcionalidad reducida (RFD) y dispositivos de funcionalidad completa (FFD). Un FFD en una red ZigBee puede actuar como coordinador, enrutador y dispositivos final, y puede construir una variedad de topologías. Mientras que un RFD solo puede comunicarse con un FFD [8].

La topología de malla es una extensión de la comunicación entre pares con las siguientes características:

- Los nodos ruteadores pueden tener hijos.
- Comunicación directa entre nodos FFD.
- Los nodos terminales solo pueden intercambiar datos con sus nodos padres.
- Es posible un ruteo dinámico.

Para el mecanismo de ruteo un algoritmo implementado en la capa de red muy popular por su simplicidad y bajo requerimiento de procesamiento es el AODV (Ad hoc On-Demand distance Vector) En AODV los nodos mantienen una tabla de ruteo para los destinos conocidos. En el comienzo esta tabla la integran sus vecinos. Solo se agrandará la tabla cuando aparezca algún nodo con camino desconocido. En este caso se envía mensajes de descubrimiento que se propagan entre los nodos hasta llegar al destino. Desde el destino se inicia el camino inverso hasta llegar al nodo origen. Todos los nodos actualizan sus tablas [8].

## Capa de aplicación

Consiste en las subcapa: *Application Support* (APS) y la *ZigBee Device Object* (ZDO). APS trata de descubrir otros dispositivos que están operando en su mismo espacio operativo y ZDO define el rol de un dispositivo dentro de la red. Es responsable de mantener las tablas para los enlaces que consiste en balancear o adaptar dos dispositivos entre ellos basados en los servicios y necesidades. En esta capa se inician o responden pedidos de enlace y se

establece una relación segura entre dispositivos seleccionando un método de seguridad como una clave. ZigBee define puntos finales de aplicaciones que son los destinos, estos actúan como puertos de comunicación. Un nodo puede tener hasta 240 puntos finales de aplicación [9].

## 6.6. Fallas en la red ZigBee

En una red robusta, la falla de un dispositivo no debe terminar en la falla de la red completa. Sin embargo, con el estándar de ZigBee, una falla del coordinador puede terminar en la falla completa de la red para una topología de estrella o en una falla parcial en una topología de árbol. La comunicación también puede fallar independientemente de la topología seleccionada si se implementa una vinculación indirecta porque la tabla de vinculación describe la información de ruteo necesaria y está almacenada en el coordinador. Se pueden identificar dos tipos de de impacto por las fallas: uno debido por la estructura de la red, mientras que el otro por las características que ZigBee utiliza [10].

El impacto por fallas en la estructura parte de las topologías disponibles en ZigBee: estrella, árbol o malla. La falla de los dispositivos tienen un impacto distinto en cada topología. Cuando un dispositivo falla es incapaz de comunicarse con los otros. Para una topología de estrella, el dispositivo central se comunica con todos los demás. La comunicación completa se detiene si dispositivo central llega a fallar. Este es el peor escenario para una falla si se trata de falla de un punto singular en un nodo. En una topología de estrella el coordinador siempre es el dispositivo central y por ello se le dota de cierta importancia [10].

La segunda topología es la de árbol con nodos como ramas y hojas de un árbol. Si una hoja falla, es incapaz de comunicarse, pero todos los demás dispositivos pueden comunicarse con normalidad. No obstante, la falla de un nodo interno o una rama separa la red en varias partes. Esta situación puede solucionarse si la red cuenta con los servicios para reconectar el árbol. ZigBee incluye esta funcionalidad, pero se deben de tomar bastantes consideraciones ya que las direcciones cambiarán y deben de existir suficiente nodos libres. La raíz del árbol tiene un rol especial, este nodo siempre es el coordinador. Si el coordinador falla, el subárbol tratará de conectarse a un nodo raíz. Sin embargo, como fallo el coordinador y al ser el único coordinador en la red, el sub-árbol no puede conectarse y la red permanece únicamente en el sub-árbol o, en el peor de los casos, falla la red completa [10].

La última topología y la más robusta es una topología de malla. Como en la de árbol, los dispositivos pueden ser nodos internos u hojas. Si embargo, la diferencia radican en que todos los nodos son iguales y pueden comunicarse con todos los nodos que están en el rango de comunicación. Esto le permite a la red a auto-repararse al buscar una ruta alternativa para la comunicación, siempre y cuando exista más de un nodo en el rango.

Otra tipo de falla independiente de la topología seleccionada es por falla del coordinador. ZigBee cuenta con tipo especial de comunicación llamado *binding*. *Binding* es utilizado para modificar la red en la inicialización. Existen dos tipos *binding*: *direct binding* y *indirect binding*. En el primero los dispositivos que se comunican almacenan la información de envío internamente. Por ello, la comunicación solo falla si no es posible encontrar una ruta entre los dos dispositivos. *Indirect binding* significa que toda la información para el enlace es

almacenada en el coordinador y, los dispositivos que quieren comunicarse envían una paquete al coordinador que luego es dirigido al destinatario. En este caso, el coordinador se vuelve un punto singular de falla: cuando falla, independientemente de la topología utilizada la red falla [10].

## 6.7. Protocolo 802.15.4

Entre las características más importantes que se pueden mencionar son[9]:

- Trabajar en bandas de 2.4 GHz como en 868MHz o 915MHz.
- Tasa de transmisión de 250 Kbps a 2.4 GHz, 40Kbps a 915MHz y 80 Kbps a 868 MHz.
- Optimizado para ciclo efectivo menor a 0.1 por ciento.
- Usa CSMA-CA para acceso al canal. Cada equipo anuncia su intención de transmitir antes de hacerlo para evitar colisiones entre los paquetes de datos.
- Baja potencia.
- Produce alto rendimiento y baja latencia para dispositivos de bajo ciclo de trabajo.
- 64 bits de direccionamiento determina una cantidad máxima de  $1.8 \times 10^{19}$  dispositivos.
- 16 bits para identificar redes que determina un total de 65536 redes.
- Permite ranuras de tiempo para aplicaciones de baja latencia.
- Protocolo con handshake para mejorar la seguridad en las transferencias.
- Rango: 50 m.

Para las distintas aplicaciones de ZigBee se tiene un tráfico que puede clasificarse en uno de los siguientes tipos:

- Datos periódicos (continuo): La aplicación define una tasa de datos. Es un caso típico de sensores en donde por ejemplo un sensor necesita transmitir la temperatura cada 10 segundos.
- Datos intermitentes (por evento): En este caso la aplicación junto a otros estímulos externos al dispositivo definen la tasa de datos. Por ejemplo, la transmisión ocurre hasta oprimir un interruptor.
- Datos periódicos con comunicación garantizada (GTS): (Guaranteed time slot) Es para aplicaciones de baja latencia en la que se requiere comunicación libre de competencia por el canal [9].

## 6.8. *Broadcast*

En el manual del módulo MRF24J40 en la sección de recepción [11], se reconoce como un mensaje válido un mensaje de *broadcast* cuando la red y la dirección del destino son 0xFFFF y 0xFFFF, respectivamente. Esta información será útil para desarrollar los servicios relacionados a la asociación de una red.

## 6.9. Estudio de compensación

Un estudio de compensación es una actividad de toma de decisión utilizada para identificar la solución técnica más aceptable de una serie de propuestas. Por naturaleza, todas las decisiones son subjetivos e involucran un grado de riesgo. Un estudio de compensación provee un medio efectivo que aborda la toma de decisiones al documentar el proceso para permitir trazabilidad y repetitividad.

Soluciones potenciales de un estudio de compensación son juzgadas por el conjunto de satisfacción para una serie de características deseables. Las características pueden entrar en conflicto entre ellas o ser mutuamente excluyentes [12].

## 6.10. Métodos de sincronización en redes inalámbricas de sensores

Una WSNs, por sus siglas en inglés, consiste en un conjunto de sensores distribuidos en un espacio que interactúan unos con otros para formar una red. Estas redes son robustas, adaptables al cambio y fáciles de programar. Todos los nodos cuentan con el *hardware* de un reloj que son configurados a un tiempo en común, pero los relojes pueden variar en la frecuencia y la fase por aspectos físicos como la temperatura, humedad, voltaje, antigüedad, proceso de manufactura, entre otros. La sincronización de los relojes es esencial para mapear eventos en tiempo real, agendar algoritmos y evitar colisión para una transmisión libre. Por este motivo, las técnicas o los métodos para este tipo de redes pueden ser de gran beneficio para redes enfocadas en aplicaciones de robótica de enjambre [13].

Las principales fuentes en la latencia de los mensajes en la sincronización de tiempo son [14]:

- Tiempo de envío: El tiempo que le toma el emisor construir el mensaje.
- Tiempo de acceso: En la transmisión de paquetes se enfrenta a un *delay* en la capa MAC. Este *delay* depende del tiempo en que el canal se encuentre libre.
- Tiempo de propagación: Es el tiempo que se requiere enviar el mensaje desde el emisor hasta el receptor.
- Tiempo de recepción: Es el tiempo necesario por el receptor para recibir el mensaje del canal y notificar a las capas superiores del nuevo mensaje.

El motivo de necesitar la sincronización en una red son [14]:

- Los nodos requieren de coordinar su operación para desarrollar cierta tarea.
- El tiempo de vida de una red depende de la energía.
- Para compartir en el medio de transmisión se requieren de algoritmos para sincronizar mensajes.

Los principales cuatro protocolos para sincronización en una red WSNs se describen a continuación.

### 6.10.1. *Reference broadcast synchronization*

El protocolo RBS, por sus siglas en inglés, es una sincronización de receptor a receptor. Un nodo manda una baliza por *broadcast* a todos los receptores. Los receptores comparan sus relojes con los de los demás para calcular su desfase relativo y transmitir el mensaje. El tiempo de referencia está basado en cuando los nodos reciben la baliza. La forma de operación es la siguiente. El paquete de tiempo será enviado a los receptores y ellos almacenarán el tiempo cuando recibieron el paquete. Luego comparten entre los receptores la información de sus tiempos de recepción y calculan el desfase que tienen [14].

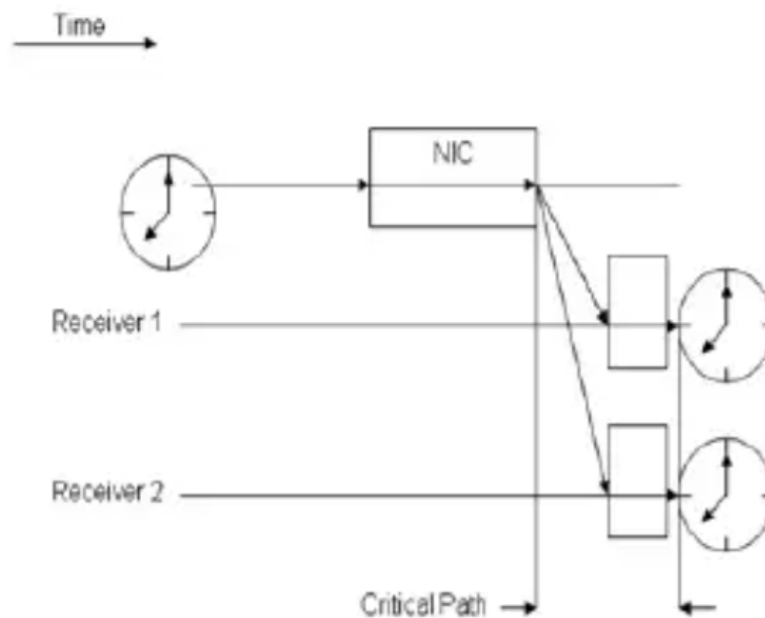


Figura 6: Sistema de sincronización con RBS.

Las ventajas de esta red son:

- Elimina el emisor del camino crítico.

- El tiempo de propagación es despreciable en redes donde el rango es relativamente pequeño.
- Al remover el emisor y el tiempo de propagación la única fuente de error es el tiempo de recepción.

### **6.10.2. *Time sync protocol for sensor networks***

El protocolo TPSN, por sus siglas en inglés, es aplicado en redes en forma de árbol. Está constituido por dos niveles; el nivel de descubrimiento donde cada nodo determina su nivel en la red. El segundo nivel se llama nivel de sincronización, y este comienza en el nodo raíz y se propaga a los otros niveles. Aquí, todos los nodos en el nivel 'i' se sincronizan con los nodos del nivel 'i-1'. Primero, el nodo raíz hace la asignación. Si un nodo está equipado con receptor GPS, entonces ese debería ser el nodo raíz y todos sincronizarse al tiempo global [14]. Las ventajas son:

- Minimiza el tiempo en la transmisión.
- No hay problemas relacionados al rango ya que está enfocado a una red multi saltos.
- Tiene una mejor precisión que RBS y la sincronización entre emisor y receptor es superiora la sincronización de receptor a receptor.

### **6.10.3. *Flooded time synchronization protocol***

El protocolo FTSP, por sus siglas en inglés, es una sincronización entre emisor y receptor. Este protocolo es similar al TPSN, pero con mejoras en las desventajas del otro. Es similar en el hecho que tiene una estructura de un nodo raíz y que los nodos se sincronizan respecto a la raíz. En este protocolo los nodos pueden formar una malla donde cada nodo envía un mensaje de sincronización a los demás nodos, o pueden formar una red en estrella donde solo un nodo actúa como coordinador. El coordinador envía periódicamente un mensaje de sincronización. Todos los nodos en la red reciben el mensaje y sincronización su reloj interno al del coordinador [14]. Las ventajas son:

- FTSP es robusto en la sincronización de datos y el fallo de un nodo.
- Provee la habilidad de cambiar dinámicamente la topología.
- Elimina la mayoría de fuentes de latencia a excepción del tiempo de propagación.
- Puede utilizar una regresión lineal para estimar el desfase en su reloj.

### **6.10.4. *Time Synchronize mesh protocol***

El protocolo TSMP, por sus siglas en inglés, brinda una segura transferencia al combinar tiempo, frecuencia y canales. Todos los nodos comparten la misma noción de tiempo,

certeza de  $< 1$  ms. Esta capacidad permite diversificar diversificación de tiempo al agendar distintas transmisiones para que ocurran durante intervalos donde no se traslapen y de esta manera evitar la colisión de datos. Cuando los nodos están de acuerdo con cuándo comunicarse, ambas partes (receptor y emisor) pueden reducir su tiempo de la antena encendida. Al añadir los saltos de canal, el protocolo ofrece ventajas de la diversidad de frecuencias. Primero, incrementa de forma lineal el ancho de banda con la cantidad de canales disponibles. Segundo, incrementa la fiabilidad de la red ya que existen más canales por los cuales se pueden comunicar [15].

## 6.11. Algoritmos de elección de coordinador

Uno de los principales problemas en la transición de una simulación a una implementación física es mantener los mismos aspectos que lo describen. En la robótica de enjambre, el aspecto de robustez está directamente conectado con el punto único de fallo. Las fallas de punto único son aquellas en las que el fallo de un sistema ocasiona el fallo del sistema completo. Para topologías centralizadas como estrella o árbol, la falla del nodo central significa que la red completa falle. Por lo tanto, los algoritmos de elección de coordinador resuelven o mitigan este problema [16].

Varios algoritmos requieren la elección de un coordinador para los procesos; un sistema que tenga un rol especial o que inicie un proceso o que esté monitoreando algo. Normalmente no se requiera que tenga una cualidad especial sobre los demás, pero se requiere que exista un coordinador. Por lo tanto, para redes descentralizadas existen dos algoritmos básicos: algoritmo bravucón y algoritmo de anillo.

### 6.11.1. Algoritmo bravucón

Los supuestos que tiene este algoritmo son [17]:

- El sistema esta sincronizado: utiliza interrupciones de que se acabó el tiempo para detectar la falla del proceso.
- Cada proceso en el sistema tiene un número de proceso único.
- Todos los procesos conoce el número de proceso de todos los procesos y sabe diferenciar entre los que tienen un número mayor.
- Cuando una elección ocurre, el proceso con el número de proceso más grande es elegido como coordinador.
- En la recuperación, un proceso que falló puede tomar intentar unirse nuevamente.

Existen tres tipos de mensaje en este algoritmo. El de elección que es enviado para anunciar una elección, una respuesta al mensaje de elección y un mensaje de coordinador es enviado para anunciar el número de proceso elegido. El proceso de elección inicia cuando se nota a través de la interrupción de tiempo que el coordinador fallo. El proceso que sabe

que tiene el número más grande se auto elige como coordinador simplemente enviando un mensaje de coordinador a todos los procesos con número menor. Por el otro lado, un proceso con un número menor comienza una elección al enviar un mensaje de elección a los procesos que tienen un número más grande y espera una respuesta de regreso. Si ninguno respondió y ocurre la interrupción, el proceso se considera el coordinador y envía un mensaje de coordinador a todos los procesos con un número menor [17].

### 6.11.2. Algoritmo de anillo

Este algoritmo es similar al algoritmo bravucón, pero los nodos o procesos son ordenados de una manera lógica en anillo, y los nodos solo se comunican con sus vecinos lógicos. Cuando el líder se pierde, los vecinos lo notaran. La fase de negociación es alcanzada por los nodos pasando el mensaje alrededor del anillo para establecer quien es el nodo con el número más grande y llegan a un consenso [18].

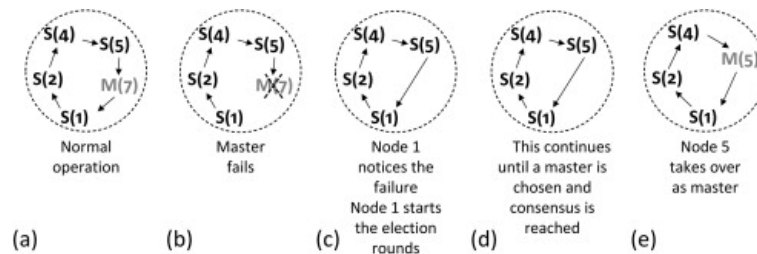


Figura 7: Operación del algoritmo de anillo [18]

En la Figura 7 se muestra la operación del algoritmo de anillo. En la parte (a), se muestra el escenario normal de un único líder; los nodos están conectados en una forma lógica de anillo que es mantenida por mensajes periódicos de un nodo a otro. El mensaje informa al receptor que el nodo vecino siguen conectados. En la parte (b), el líder falla, lo cual hace que se abra el anillo. En la parte (c), el nodo 1 notifica la ausencia del líder e inicia la elección al enviar un mensaje especial a su vecino lógico (nodo 2). En la parte (d), la elección se propaga alrededor del anillo hasta que el nodo con el número más grande es identificado y todos los nodos están al tanto de su presencia. En la parte (e), el nodo 5 se declara como líder [18].

### 6.11.3. Algoritmo adicional

Además de los algoritmos básicos para la elección de un coordinador, existe un grupo de algoritmos conocidos como algoritmos de consenso. Estos algoritmos son utilizados primordialmente en el entorno de criptomonedas. Estos algoritmos permiten a sistemas distribuidos a trabajar en conjunto y mantener seguridad. Los algoritmos de consenso en sistema de criptomoneda han ayudado a prevenir cierto tipos de ataques. En teoría, un atacante podría comprometer estos algoritmos si controla el 51% de la red. Algunos de los más conocidos son: *Proof of Work*, *Proof of Stake*, *Byzantine Fault Tolerance*, *Paxos*, entre otros [19]. Para el caso de elección de coordinador se tiene *Raft*.

*Raft* es un algoritmo de consenso que está subdividido en tres partes: elección de líder, replicación de registros y seguridad. Para la aplicación de robótica de enjambre la única parte de interés es la elección de líder [20].

La elección utiliza un mecanismo de latido para activar la elección de líder. Cuando los sistemas inician, comienzan como seguidores. Un sistema permanece como seguidor mientras continúe recibiendo latidos del líder. El líder envía periódicamente latidos a todos los sistemas para mantener la autoridad. Si un seguidor no recibe un latido en un periodo de tiempo definido como “tiempo de elección”, entonces asume que no hay un líder viable y comienza el proceso de un nuevo líder.

Para iniciar la elección, el seguidor cambia como un potencial candidato. Después vota por si mismo y manda una solicitud de votos a todos los sistemas. Un candidato continua hasta que una de tres razones ocurre: gana la elección, otro sistema se establece como líder o pasa el “tiempo de elección” y no hubo ningún ganador.

Un candidato gana la elección si recibe la mayoría de los votos de los sistemas. Cada sistema votará por un candidato basado en una condición, el primero en solicitar el voto. La regla de la mayoría asegura que solo un candidato puede ganar para un período específico. Una vez el candidato gana la elección, se convierte en el líder. Luego envía su latido a todos los sistemas.

Mientras espera los votos, el candidato puede recibir latidos de otro sistema diciendo que es el líder. Si el número del líder es mayor a la del candidato, el candidato acepta al nuevo líder. Si es menor, lo rechaza y sigue siendo un candidato. La tercera posibilidad es que ningún candidato gane. Cuando esto sucede se reinicia el proceso de elección [20].

## 6.12. Rendimiento en una red

### 6.12.1. Rendimiento en redes descentralizadas

El rendimiento de una red se puede medir como la cantidad de data transmitida restringida por un conjunto de recursos. En el artículo de D’Souza, Ramanathan & Lamb [21] definen que se puede medir indirectamente en términos de tres métricas: consumo de energía, interferencia de otros usuarios y el tiempo de transmisión de un mensaje. Estas métricas son cantidades que pueden medirse y son naturalmente estadísticas y la distribución de ellas dependen de propiedades propias de la red como técnicas de ruteo, topología, densidad de usuarios, entre otras.

Sin embargo, en el artículo de Jyothi, Singla, Godfrey & Kolla [22] proponen dos formas de medir el rendimiento de una red en base de su topología. La primera es estudiar las matrices de tráfico experimentalmente medidas. La segunda es generar la matriz de tráfico para el peor escenario con un algoritmo propuesto en el artículo. El motivo de medir el rendimiento utilizando las matrices de tráfico es que proporcionan información acerca del escalamiento, robustez y el efecto de una colocación esparcida.

### 6.12.2. Rendimiento en la red Zigbee

La tesis de maestría de Kui Liu [23] realiza una evaluación del desempeño de una red Zigbee en un ambiente real. Construyó un marco de pruebas con el fin de evaluar el desempeño de Zigbee para aplicaciones de vatíhorímetros o contadores eléctricos y las métricas estudiadas incluyen:

- tasa de rendimiento efectiva
- Tamaño de la red
- Seguridad
- Consumo energético
- Alcance y retardo en paquetes

En el caso de la seguridad, ZigBee ofrece ocho niveles de seguridad para asegurar la integridad y confidencialidad de los datos. Además, el tamaño de la red para Zigbee tiene como límite un poco más de sesenta y cinco mil nodos en la misma red. Para realizar los cálculos y mediciones se encuentran las siguientes implementaciones y preparaciones.

#### Tiempo de ida y vuelta

El tiempo de ida y vuelta (RTT, por sus siglas en inglés) se calcula en la capa de aplicación. Debido a que los ACK son enviados y procesados por la capa MAC la capa APP no estará al tanto de estos mensajes. En este caso, RTT se calcula desde que se envía el mensaje hasta que reciba una respuesta.

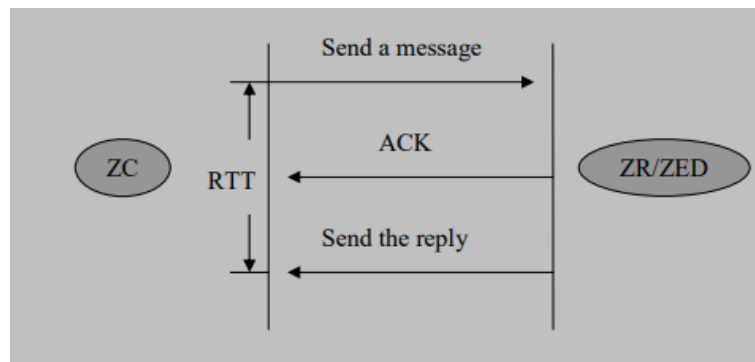


Figura 8: Un paquete de ida y vuelta

Según la Figura 8, RTT es calculado desde que un paquete es generado en la capa APP y enviado a capas inferiores hasta el tiempo en que la capa APP recibe una respuesta del receptor. Consiste de cuatro partes:

- Tiempo de procesado, agregar y remover encabezados

- Tiempo de cola, en una transmisión indirecta, el paquete será manejado por un nodo extra antes de llegar al destino
- Tiempo de transmisión a una velocidad de transmisión fija, 250kpbs por ser Zigbee
- Tiempo de propagación, propagación por el aire

Si un paquete es perdido durante la transmisión, una retransmisión también contribuye al RTT completo.

La retransmisión es manejada por la capa APP, la capa NWK y la capa MAC. Si un reconocimiento está habilitado en la capa APP, esta capa realizará unos cuantos intentos antes de recibir el ACK. La capa NWK también ejecuta un número de intentos para el siguiente mensaje, una vez que la capa MAC agota los intentos para el mensaje. Los ACK de la capa MAC y los intentos están predeterminados y son automáticos independientemente del servicio que se esté usando. La capa MAC realiza tres intentos, configurable, antes de regresar un estatus de error a la capa NWK. El intervalo de tiempo entre cada intento depende del CSMA/CA [23].

## Implementación

En la aplicación, la medición comienza al generar un mensaje *broadcast* desde el coordinador. Cuando los demás nodos reciben el mensaje, le responden con su dirección única; los dispositivos entonces son enumerados acorde a un índice según como fueron respondiendo al coordinador. El coordinador almacena los índices y las direcciones en una tabla.

Nuevos dispositivos se pueden unir a la red al enviar un mensaje de *broadcast*. Cuando el coordinador recibe un mensaje de un dispositivo, revisa si el dispositivo ya pertenece a la red. De ya pertenecer, le envía la información del número asignado que tiene asignado y el rol que cumple. De no pertenecer, el coordinador agrega al nuevo dispositivo a la red.

En la aplicación, los paquetes son generados en base a los ajustes establecidos y transmite con una ventana de tamaño uno, lo cual significa que el emisor no enviará un segundo paquete hasta recibir un ACK del receptor. Un temporizador es establecido para cada paquete. Si el emisor recibe el ACK antes de la alarma del temporizador, el tiempo total de ida y vuelta es calculado acorde al reloj del sistema y se informa el tiempo. Si la alarma ocurre sin que llegue el paquete, este se considera como perdido.

El resultado es procesado para revisar las condiciones de transmisión, incluyendo la tasa de paquetes perdidos y el RTT promedio. Adicionalmente, se pueden agregar características como seguridad en la integridad del mensaje y ahorro de energía al deshabilitar momentáneamente el transceptor.

Cuando los ajustes fueron establecidos, se comenzó a tomar medidas y a capturar el tráfico en la red para monitorear el proceso completo. El nivel de energía de cada paquete también fue guardado [23].

## Factores para el rango de transmisión

Las ondas de radio se propagan en línea recta en varias direcciones al mismo tiempo. En un espacio al vacío, las ondas de radio se propagan a  $3 * 10^8 m/s$ . La señal se debilita debido a:

- Reflexión
- Refracción
- Difracción
- Absorción

En el aire, la intensidad de la señal se atenúa conforme viaja debido a estas razones. La intensidad de la señal decreciente cuando la distancia incrementa. Sin embargo, no decreciente de forma lineal ya que el entorno cambia y la frecuencia cambia.

Algunos factores que afectan la radio transmisión:

- Transmisión de energía: Es un factor determinante para el rango de transmisión. Se mide en dBm y representa la intensidad de la energía que se transmite en un canal, para el caso de ZigBee un canal de la frecuencia de 2.4 GHz.
- Sensibilidad del receptor: Una sensibilidad más alta significa un mayor rango.
- Diseño de la antena: El tipo de antena y su altura afectan el rango de transmisión.

## Medición del consumo energético

Las actividades del sistema son monitoreadas después que se termine cada tarea. La evaluación que se realizó para el consumo energético utilizó la Figura 9 donde se mide el diferencial de potencial en la resistencia de  $10 \Omega$  y este potencial se divide dentro de  $10 \Omega$  para encontrar la corriente consumida por el circuito [23].

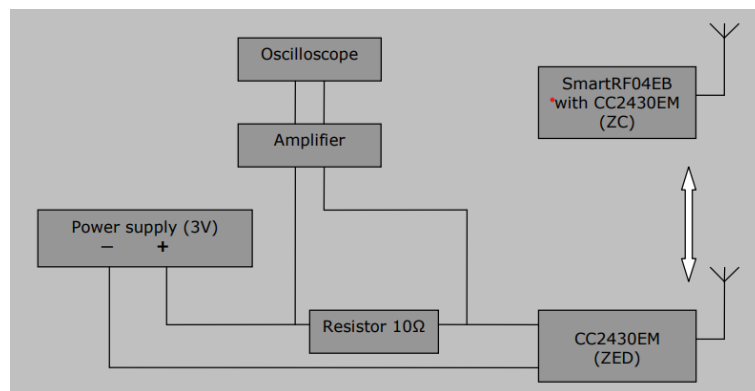


Figura 9: Circuito para el consumo energético [23].

## Tasa de rendimiento efectiva

ZigBee tiene una transferencia de datos igual a 250kbps para la banda de 2.4 GHz. Sin embargo, en la realidad, la tasa efectiva de transferencia es menor debido a los encabezados del protocolo. La prueba para la tasa de transferencia efectiva consiste en implementar una aplicación para medir la tasa de transferencia actual. En esta prueba, el emisor transmite inmediatamente cuando recibe una confirmación, ACK, la cual es recibida como una confirmación que se envió un paquete. Cuando el receptor recibe el paquete, revisa si el tamaño y el contenido del paquete son los correctos o no. En el caso que no es el paquete correcto, este es descartado y no se utiliza para el cálculo de *bytes* recibidos [23].

---

## Elección de topología, tecnología y *hardware*

---

En este capítulo se muestra la elección de la topología, el protocolo y el hardware que son los más aptos para aplicaciones de robótica de enjambre. Como criterio principal se utilizaron las características de robótica de enjambre descritas por el paper de Group Ecole Polytechnique Fédérale (sec. 2.1). La forma en que se organizó y se tomó una decisión fue a partir de un estudio de compensación (*trade study*).

### 7.1. Criterios de elección

Para la elección de la topología y tecnología se utilizaron las características de robótica de enjambre como criterios. Se utilizaron estos criterios para validar que conceptualmente y físicamente la red construida esté orientada hacia aplicaciones de robótica de enjambre. Las características de robótica de enjambre son: auto-organización, escalabilidad y robustez [1].

La auto-organización en el sistema de comunicación debe de cumplir con los principios de retroalimentación positiva, retroalimentación negativa, aleatoriedad y múltiples interacciones. La escalabilidad radica en la capacidad que tiene el sistema para crecer sin que su rendimiento se vea afectado. La robustez es importante en la robótica de enjambre porque el comportamiento de estos algoritmos radica en la multitud y si llegara a fallar algún aspecto en la red de tal manera que se pierde la naturaleza de enjambre desaparece la inteligencia de enjambre.

## 7.2. Topología

### 7.2.1. Aspectos cualitativos de una topología

Se identificaron ocho aspectos cualitativos que describen una topología y se clasificaron en cada una de las tres características de robótica de enjambre.

- Alcance: Es la distancia que puede llegar un mensaje del punto A al punto B. Algunas técnicas empleadas en algunas topologías descentralizadas son los multisaltos permitiendo tener un mayor alcance. Mientras que en una topología centralizada los mensajes pasan por el nodo central o concentrador y el alcance depende la antena que tiene. Este aspecto se clasifica en escalabilidad.
- Tolerante a colisiones: La transmisión de datos puede verse comprometida por la colisión de paquetes. La colisión de datos ocurre cuando dos o más nodos detectan transmiten al mismo tiempo y aumenta cuando existen demasiados nodos en la red. Segmentar la red, dividiéndola en grupos unidas por un puente o conmutador puede reducir la colisión y algoritmos como *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD, por sus siglas en inglés) evitan o reducen la colisión de datos [24]. Este aspecto se clasifica en escalabilidad y robustez.
- Fiabilidad: La red se espera que funcione para lo que fue diseñada durante un tiempo determinado. Sin embargo, dependiendo de la falla, la red podría fallar completamente. Una de las fallas más críticas es punto de fallo único (SPOF, por sus siglas en inglés). Un SPOF es cuando en una red existen partes no redundantes que causan el fallo completo de la red, como un ruteador en una red WiFi. Cuando los nodos son redundantes, es decir, no presentan ningún riesgo al funcionamiento de la red como los nodos finales en una red, se pueden tener fallos sin afectar la red [25]. Este aspecto se clasifica en robustez.
- Complejidad: la dificultad que representa utilizar la topología e instalarla. Una red resulta más compleja cuando se debe de tener una infraestructura muy costosa y que requiera de herramientas muy especializadas. Este aspecto se clasifica en robustez y escalabilidad.
- Flexibilidad: La capacidad que tiene la topología para ser modificada. Una red que permite agregar y retirar nodos sin muchas complicaciones resulta ser más flexible. Además, que puedan implementarse más protocolos sin que la topología sea un obstáculo. Este aspecto se clasifica en auto-organización y escalabilidad [26].
- Velocidad: El tiempo necesario para que un mensaje llegue a todos los miembros de la red. Una red que presente más nodos y no se pueda tener una conexión directa implica un mayor tiempo para que llegue el mensaje y una menor velocidad. Este aspecto se clasifica en escalabilidad y auto-organización.
- Seguridad: La privacidad que se tiene entre el emisor y el receptor. Un sistema es más seguro si el mensaje es directo y es menos seguro si debe pasar por varios nodos hasta llegar al receptor. Este aspecto se clasifica en robustez.

- Enrutamiento: La transmisión de mensajes puede volverse eficiente cuando se encuentra un camino más corto o más libre entre el emisor y el receptor. En topologías que presentan redundancia se pueden tener varias vías para que el paquete llegue a su destino, esto permite tener alternativas si se presentan rutas bloqueadas. Este aspecto se clasifica en escalabilidad y auto-organización [27].

### 7.2.2. Topologías evaluadas

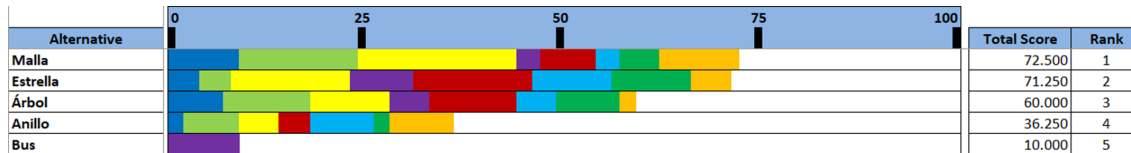
Las topologías propuestas abarcan tanto centralizadas como descentralizadas para discriminar los aspectos descritos según los criterios propuestos de robótica de enjambre. Las cinco topologías propuestas son las siguientes:

- Bus: La transmisión de datos ocurre en una dirección ya que todos los nodos están conectados por un único cable. La ventaja de esta topología es su relación costo-efectividad, requiere menos trabajo comprenderla y es fácil escalar la red. Las desventajas que presenta es punto de fallo único ya que al fallar el cable toda la red falla y el tráfico que se genera al incrementar el número de nodos en la red causando que el desempeño disminuya [27].
- Anillo: Su nombre proviene de la forma en que se está interconectada la red, Cada uno de los nodos está conectado con dos vecinos. En esta topología, si el emisor envía un paquete al receptor entonces el paquete debe de pasar por todos los nodos entre ellos. Entre las ventajas se tiene que, la transmisión de datos no es afectado por el tráfico en la red o por añadir más nodos y tiene un menor costo comparada con otras redes. Las desventajas que presenta son, la dificultad de detectar el fallo en la red y el riesgo que presenta añadir un nodo extra porque puede abrirse el anillo si falla un nodo [27].
- Estrella: En esta topología, todos los nodos conectados están conectados a un único nodo central llamado *hub*. Cada dispositivo tiene su propia conexión al nodo central. Adicionalmente, el dispositivo central actúa como repetidor de los mensajes que recibe. Las ventajas que tiene son, un mejor desempeño comparado con otro tipo de redes. Además, el mantenimiento, la instalación y las modificaciones de los nodos en la red no tienen ningún efecto. Sin embargo, la principal desventaja es que cualquier falla del nodo central significa una falla completa de la red. También, el costo de instalación de esta red tiene un mayor costo que las redes descritas anteriormente [27].
- Árbol: Existe jerarquía en este tipo de topología, un nodo en esta red debe ser la raíz del que parten todos los demás nodos. Esta topología se considera una extensión de otras como de bus y estrella, agregar nuevos nodos es sencillo y tiene la capacidad de detectar errores. Entre las desventajas que tiene es que presenta un mayor costo y la más crítica para robótica de enjambre es el fallo completo de la red si se pierde el nodo central [27].
- Malla: En esta topología existe una comunicación entre pares y tiene  $n(n - 1)/2$  canales físicos para conectar  $n$  nodos. En consecuencia, se tienen dos técnicas para la transmisión de datos: enrutamiento e inundación. El enrutamiento es la capacidad de mover paquetes a través de la red, se encuentra una ruta entre el emisor y el receptor para enviar un paquete. El algoritmo de inundación se trata de un método en el que

al recibir un paquete, el nodo lo retransmite por todos sus enlaces, excepto por el que llegó el paquete. Esta topología es robusta, pero resulta complicado su instalación y configuración [27].

### 7.2.3. Elección de la topología para robótica de enjambre

Se realizó un estudio de compensación en el cual se midió cada propuesta en los ocho aspectos que se describieron anteriormente y se utilizaron [26], [27], [28], [29] y [30] para calificar cada topología. Para cada aspecto se calificaron las topologías de mejor a peor siendo 1 la peor topología y 5 la mejor topología. Finalmente, a cada aspecto se le asignó un peso bajo el criterio de las características de robótica de enjambre donde un peso igual a 10 es el peso base. La topología que tuvo un mejor resultado para las aplicaciones de robótica de enjambre es la topología de malla como se observa en la Figura 10.



Criteria	Weight
Alcance	10.000
Tolerante a colisiones	15.000
Fiabilidad	20.000
Complejidad	10.000
Flexibilidad	15.000
Velocidad	10.000
Seguridad	10.000
Enrutamiento	10.000

Figura 10: Estudio de compensación de la topología.

Se asignó a la fiabilidad un mayor peso bajo el criterio de robustez porque el funcionamiento correcto de los algoritmos de enjambre implican una comunicación segura entre los robots del enjambre. Si llegara a caerse la red, el enjambre completo dejaría de funcionar.

En segundo lugar, se les asignó un mayor peso a los aspectos tolerante a colisiones y flexibilidad; el primero bajo los criterios de robustez y escalabilidad, mientras que el segundo bajo los criterios de escalabilidad y autoorganización. Una red debe ser tolerante a colisiones ya que la robótica de enjambre implica tener un grupo de robots que puede escalar en gran número de robots y de no ser tolerante a colisiones se pierde el comportamiento colectivo del enjambre. La robótica de enjambre debe ser flexible ya que para que los robots cumplan con sus tareas deben de posicionarse de manera dinámica y este proceso requerirá de varios algoritmos, protocolos y servicios de la red. Por lo tanto, una mayor flexibilidad en la red brindará una mayor libertad al enjambre.

En tercer lugar está el enrutamiento bajo los criterios de autoorganización y escalabilidad. Cualquier mejora en la transmisión de los paquetes significa que los servicios de la red fluyen

con mayor facilidad y mejora la comunicación para redes con varios nodos. En consecuencia, mejora el comportamiento colectivo.

Finalmente, el alcance, la complejidad, la velocidad y la seguridad el menor peso porque si bien afectan la red en general y tendrán un impacto para la robótica de enjambre, no son suficientemente significativos como los demás aspectos.

## 7.3. Tecnología

### 7.3.1. Tecnologías compatibles con la topología seleccionada

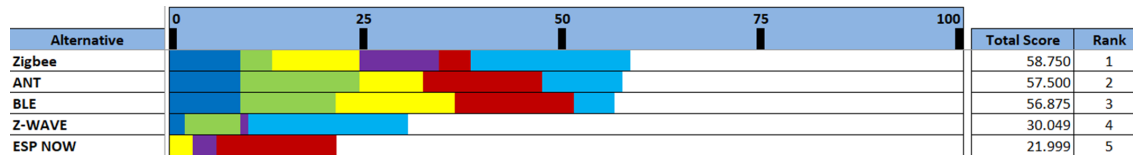
Al tener la topología seleccionada se prosiguió a deliberar entre las tecnologías de comunicación inalámbrica. Se seleccionaron las cinco opciones que pueden implementarse en una topología descentralizada y los aspectos que se utilizaron para describir las redes son más puntuales, debido a que son variables cuantitativas en su mayoría y se obtuvieron de las páginas oficiales de cada tecnología [31], [32], [33], [34], [35], [36] y [37].

Tecnología	Nodos	Consumo [mA]	Costo [\$]	Alcance [m]	Velocidad de transmisión [Mbps]	Topología
Zigbee	>65000	29	3.92	10-100	0.250	mall,estrella
BLE	>65000	18	2.38	50	1.000	p2p
ESP NOW	20	200	4.20	100	1.000	p2p, estrella
ANT	>65000	16	3.93	<30	1.000	p2p, estrella, árbol, malla
Z-WAVE	232	23	9.84	30	0.100	estrella, malla

Cuadro 2: Tecnologías para topologías descentralizadas.

### 7.3.2. Elección de la tecnología para la topología seleccionada

Al igual que con la topología se realizó un estudio de compensación para determinar la tecnología implementada en la red. Se compararon las cinco tecnologías en los aspectos descritos en el Cuadro 2 y se enumeraron de 1 a 5, siendo 1 la peor tecnología y 5 la mejor tecnología. Además, se utilizó el criterio de las características de robótica de enjambre para darle un peso a cada aspecto y, nuevamente, el peso base es igual a 10. La tecnología que tuvo un mejor resultado para aplicaciones de robótica de enjambre fue ZigBee como se observa en la siguiente Figura.



Criteria	Weight
Nodos	10.000
Consumo [mA]	15.000
Costo [\$]	15.000
Alcance [m]	10.000
Velocidad de transmisión [Mbps]	15.000
Topología	20.000

Figura 11: Estudio de compensación de las tecnologías.

Se asignó al aspecto de topología la mayor importancia en el estudio de compensación debido a que la topología que se seleccionó era una topología en malla. En consecuencia, la tecnología que se busca debe de ser compatible con esta topología. Luego, se les asignó un peso poco menor al consumo, al costo y a la velocidad de transmisión. Debido a que las aplicaciones están orientadas a robótica de enjambre, algunos algoritmos requieren de un funcionamiento autónomo y por ello se tomó en cuenta el consumo de la tecnología. También, al ser un grupo de robots el costo puede incrementar y una tecnología de bajo costo puede reducir los gastos del hardware que se necesita. Bajo el criterio de las características de robótica de enjambre una mayor velocidad de transmisión significa un mejor comportamiento del enjambre. Finalmente, como últimos aspectos se tienen el número de nodos y el alcance. El primero, ya que todas las tecnologías permiten conectar varios nodos a la red, y el segundo, porque entre los objetivos que se plantearon está diseñar e implementar el sistema de comunicación en la mesa de pruebas de UVG, todas las tecnologías cumplen con la restricción de alcance para las dimensiones de la mesa.

## 7.4. Transceptor

### 7.4.1. Propuestas compatibles con la tecnología seleccionada

Entre los dispositivos compatibles con esta tecnología se seleccionaron dos por su fácil implementación y bajo costo relativo a los demás módulos de la misma tecnología. El primero es *Xbee Module ZB series S2C* [38] que tiene un consumo de 40 mA mientras está operando y menos de 1  $\mu A$  cuando está en modo dormido y un alcance de 40 metros en lugares cerrados. El segundo es el módulo MRF24J40MA desarrollado por Microchip [11] que tiene un consumo de 23 mA mientras está operando y 2  $\mu A$  mientras está dormido y un alcance de 38 metros en lugares cerrados. Ambos módulos presentan parámetros parecidos, sin embargo, la principal diferencia radica en los precios ya que el primero tiene un precio de \$22.95 y el segundo de \$7.40. Por otra parte, Xbee es un módulo del que se encuentra mayor documentación e información para implementar y programar que MRF24J40MA.

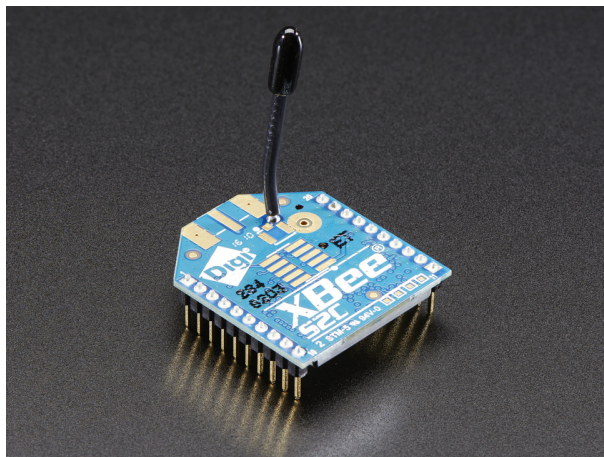


Figura 12: Módulo Xbee [38].

XBee Module - ZB Series S2C - 2mW with Wire Antenna - XB24CZ7WIT-004	
Precio unitario	\$ 22.95
Precio por 5 módulos	\$ 114.75
Courier	\$ 10.00
Proveedor	Adafruit
Courier	Aeropuerta

Cuadro 3: Cotización para el módulo Xbee.

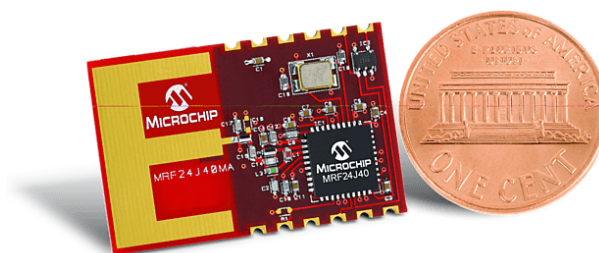


Figura 13: Módulo MRF24J40MA [11].

MRF24J40MA	
Precio unitario	\$ 7.40
Precio por 5 módulos	\$ 37.00
Courier	\$ 10.00
Proveedor	Onlinecomponents
Courier	Aeropuerta

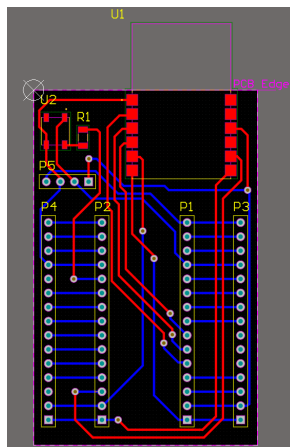
Cuadro 4: Cotización para el módulo MRF34J40MA.

### 7.4.2. Elección del transceptor compatible con la tecnología seleccionada

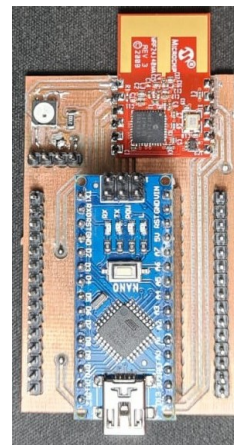
El dispositivo que se seleccionó fue el MRF24J40MA desarrollado por Microchip por su bajo costo y baja carga de procesamiento del microcontrolador a utilizar. El módulo de Microchip tiene integrado la capa física y la capa MAC por lo que se reduce el costo computacional por parte del MCU. Entre las características a destacar de este módulo están [11] :

- Detección de energía. Es útil ya que se podrá medir el nivel de energía en el canal y servirá para determinar si el canal está ocupado. Sin embargo, no puede determinar si esa energía corresponde a otro dispositivo 802.15.4 ó no.
- Escucha de señal portadora (CS, por sus siglas en inglés). Esto consiste en demodular la señal recibida para determinar si es compatible con el estándar. El canal se considera ocupado si la señal es compatible con 802.15.4.
- Evaluación de canal libre (CCA, por sus siglas en inglés) y mecanismo CSMA-CA. Este mecanismo hace que la MAC le pida a la capa física que evalúe el canal para ver si está libre. El CCA es parte del manejo de la capa física.
- Retransmisión automática de datos.
- Respuesta ACK automática.
- Transmisión independiente o con baliza.
- Encriptación y descryptación para la capa MAC y capas superiores.

Adicionalmente, se desarrolló un *shield* para el Arduino Nano [39] con el fin de realizar las pruebas del módulo y mediciones de la red. El programa para desarrollar la placa fue “Altium Designer”. El *shield* cuenta con una led RGB, el módulo MRF24J40MA y los pines machos para conectarse al Arduino Nano.



(a) Diseño de la placa



(b) Placa manufacturada

Figura 14: *Shield* desarrollado para Arduino Nano.

En este capítulo se encuentra cada uno de los servicios que se desarrolló para la red descentralizada. Primero se especificaron los supuestos de la red para su diseño. Luego, se describen las funciones básicas que se tuvieron que desarrollar para la construcción de los servicios. Finalmente, se describen los cuatro servicios que se desarrollaron: servicio de asociación, servicio de sincronización de tiempo, algoritmo de nodos conectados, y servicio de elección de coordinador.

## 8.1. Supuestos

Para el desarrollo de la red se definieron unos supuestos. Los supuestos establecidos son los siguientes:

- Todos los canales que utiliza Zigbee en la frecuencia de 2.4 GHz están libres.
- El *broadcast* ejecutado por el coordinador es recibido por todos los nodos.
- Todos los nodos son dispositivos de funcionalidad completa (FFD).
- Todos los nodos tienen el transceptor encendido.

## 8.2. Funciones básicas

Para el manejo del transceptor se desarrollo una librería orientada a objetos. Se utilizó el lenguaje de programación C++.

Se construyó una clase para el manejo del transceptor MRF24J50MA con un MCU como se observa en la Figura 15. La forma en que se comunican con el transceptor es por SPI y dos pines, uno enfocado a las interrupciones provenientes por un envío o recepción de paquete y el otro para reiniciar el transceptor. Esta información de los pines utilizados se almacenan en los atributos de la clase. Otros atributos que contiene la clases son temporizadores digitales implementados en los servicios de la librería.

Los temporizadores digitales necesitan al menos un temporizador físico para que funcionen adecuadamente. Los temporizadores digitales utilizan la interrupción generada por el temporizador físico para modificar sus contadores y activar sus banderas cuando sus contadores lleguen a cero.

En el objeto construido 15, primero se desarrollaron tres estructuras distintas para el manejo de la información: *tx\_info\_t*, *rx\_info\_t* y *pool\_t*. La primera estructura maneja la información del envío como el número de intentos de envío, si el canal está ocupado y el ACK que confirma la transmisión del paquete. La segunda estructura maneja la información recibida como la longitud del mensaje, el mensaje, la dirección del nodo emisor, la dirección de la red de la que partió el mensaje, la energía del mensaje que se recibió (RSSI) y la calidad de la conexión. La tercera estructura maneja la información de la red sobre quienes están conectados.

Para configurar el transceptor el MCU envía la información del registro que desea modificar y el valor con el que lo modificará. Para ello, se crearon dos operaciones para lectura de registros y dos funciones para escritura de registros. La razón por la que se crearon dos operaciones es porque la dirección de los registros puede ser de 8 o 12 bits, por lo tanto, una función de lectura y escritura es para direcciones de 8 bits y otra función de lectura y escritura es para direcciones de 12 bits.

Con las cuatro operaciones principales para lectura y escritura creadas se prosigue a las funciones que modifican y reciben información del transceptor por medio de registros, *Iniciar* en la Figura 15. Modificar el transceptor permite asignarle una dirección y una PAN, activar las interrupciones o dormir el transceptor, escoger el canal y la forma en que se mide el canal ocupado, configurar si es un coordinador o un nodo, configurar si se trabaja con una baliza o sin baliza, configurar el tipo de trama que acepta y activar el algoritmo CSMA-CA y activar o desactivar los ACK generados por el hardware.

La interrupción que genera el transceptor informa al MCU que ocurrió un cambio en los registros relacionados al envío y recepción de datos, es por ello que una de las operaciones de la clase maneja la información que genera la interrupción. Entonces en el evento de una interrupción por envío, la clase modifica la estructura relacionada al envío. La información que recibe puede ser un ACK que significa una transmisión correcta y una bandera que significa que el canal está ocupado. Por el otro lado, si la interrupción ocurre por una recepción se modifica la estructura relacionada a la recepción. En este caso, la estructura almacena bastante información porque almacena el mensaje completo que recibió, la longitud del mensaje, la dirección PAN de origen, la dirección de la que se originó el mensaje, la calidad del mensaje y la energía de la recepción.

Para el envío de mensajes se desarrollaron tres funciones distintas, *Comunicación* en la Figura 15: envío esperando un ACK como respuesta, envío sin esperar respuesta y broadcast.

Esta última función es vital ya que se implementa en servicios y sin ella algunos servicios se complicarían. Adicionalmente, el MRF24J40MA acepta tramas en las que se modifica el broadcast de manera que únicamente la PAN tenga la información de broadcast y la dirección de envío contenga la dirección específica del nodo al que se quiere hablar. Este aspecto permite que se transmita un mensaje a un nodo sin conocer su dirección PAN, pero sí su dirección propia. Resulta útil en servicio de asociación cuando un nodo que no pertenece a la red intente unirse. Ya que el coordinador, de la información del mensaje que recibió puede responder únicamente conociendo la dirección propia del nodo y tener la seguridad que solo el nodo que solicitó unirse recibirá el mensaje.

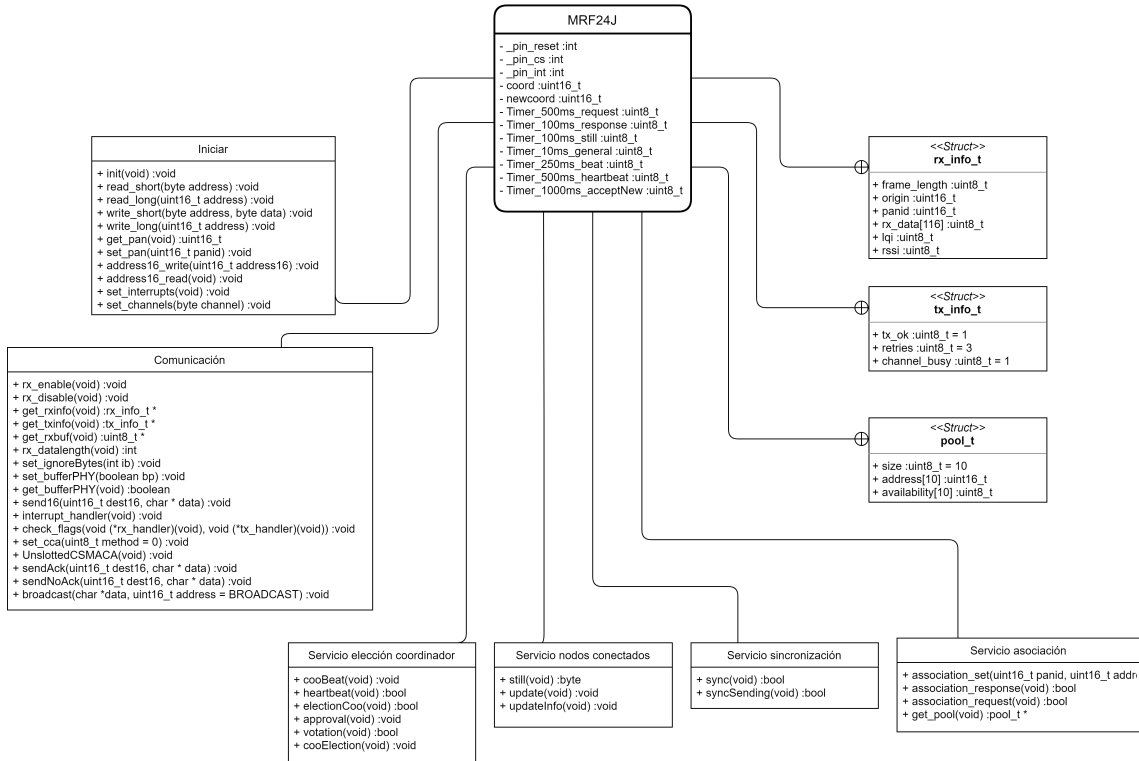


Figura 15: Diagrama de la clase MRF24J40MA.

### 8.3. Servicios

Se desarrollaron cuatro servicios para el correcto funcionamiento de la red y cumplir con las características de robótica de enjambre. El transceptor seleccionado, MRF24J40MA, incluye implementaciones y características como se mencionaron en la sección 7.4.2, que disminuyeron las funciones que fueron desarrolladas. Algunas de las implementaciones que tienen estos transceptores están: la detección de energía en el canal, escucha de señal portadora, evaluación de canal libre, mecanismo contra colisiones (CSMA-CA), retransmisión automática de datos, respuesta ACK automática, transmisión independiente o con baliza y encriptación y desencriptación. Por consiguiente, se redujo los servicios, funciones y mecanismos que se construyeron. Los servicios desarrollados se pensaron para aplicaciones de robótica de enjambre, a continuación se describen cada uno de ellos.

#### 8.3.1. Servicio de asociación

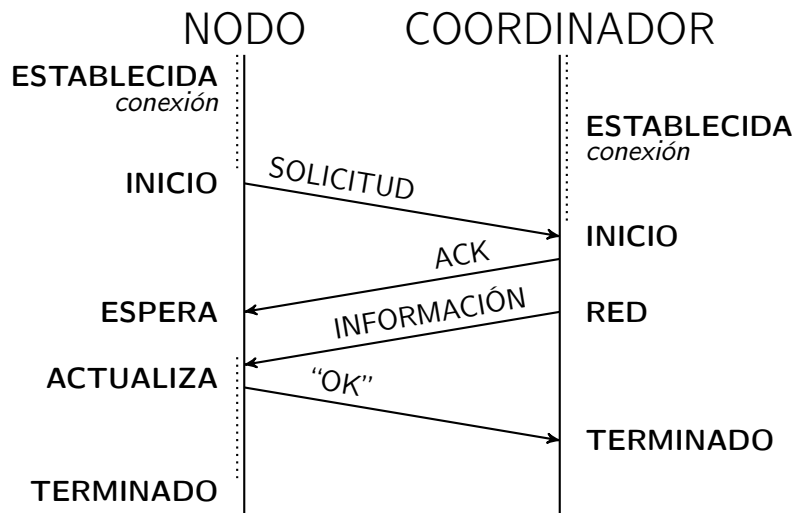


Figura 16: Secuencia de comunicación para el servicio de asociación.

Uno de los servicios necesarios en la red es la de asociación automática. Ya que en el caso que el coordinador se pierda, alguien más puede tomar su lugar y en el peor de los casos comenzar una nueva red. Entonces se desarrolló un algoritmo para que usuarios que no pertenecen a una red hagan un barrido entre los canales disponibles según el protocolo y se asocien de manera automática para que no sea necesario programarles su dirección y la red a la que pertenecen.

En la Figura 16 se encuentra la secuencia del servicio de asociación. La solicitud ocurre en la capa de Red y manda la solicitud a través de la capa inferior, MAC, para poder hacer un *broadcast* y solicitar la asociación a cualquier red existente dentro de su rango. En el instante que el coordinador de la red recibe la solicitud automáticamente responde con un *ACK* y transfiere la solicitud a la capa superior, Red, quien selecciona una dirección disponible dentro de la red para asignársela al nuevo miembro de la red. Mientras tanto, el nuevo usuario, al recibir el *ACK*, reinicia el contador para expirar el servicio y se prepara

para recibir el mensaje con la información de la nueva red. Cuando el coordinador envía la información a la dirección del usuario, este le responde con la palabra “OK” para confirmar la recepción y asociación del nuevo usuario. De parte del usuario, modifica su información en la capa de Red y manda una confirmación a la capa superior para terminar el servicio de asociación.

Se empleó un temporizador como mecanismo para determinar cuándo se acaba el servicio. Cuando ocurre la interrupción del temporizador se para el servicio y la asociación a la red fue fallida. Es importante agregar mecanismos como estos para que el MCU no quede atrapado dentro de un ciclo infinito. La implementación de los algoritmos para los nodos y para el coordinador se muestran con los diagramas de flujo de las Figuras 17 y 18.

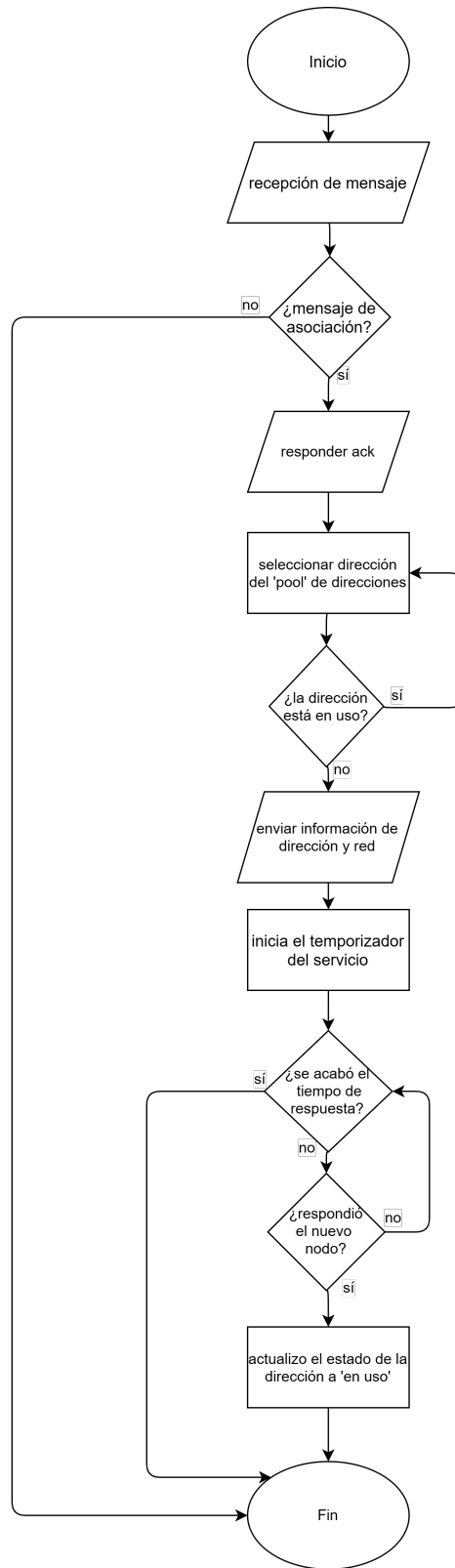


Figura 17: Algoritmo del coordinador.

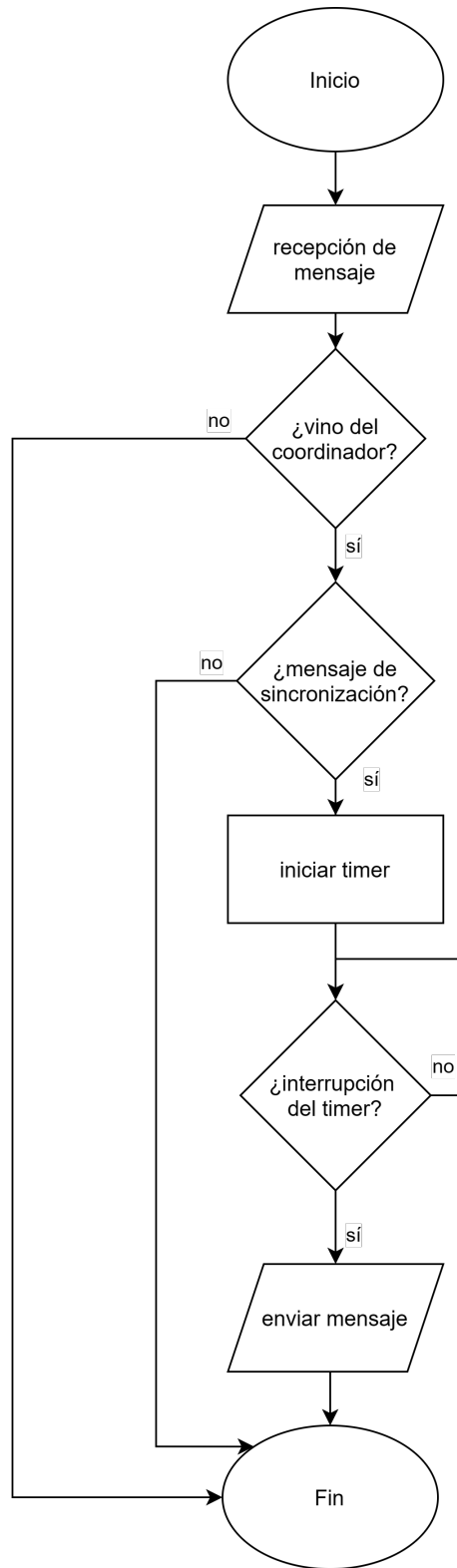


Figura 18: Algoritmo del nodo.

### 8.3.2. Servicio de sincronización de mensajes

En una red que contiene varios miembros se presentan problemas de colisión de datos. Estos aumentan cada vez que se agregan más miembros a la red. Es por ello que se han desarrollado protocolos para volver la red más robusta. Algunos protocolos para la sincronización de datos en una red Zigbee de “*Wireless sensor networks*” (WSNs) son mencionados en [40]:

- *Relative ordering*, que establece un orden relativo de mensajes y eventos al comparar los relojes locales.
- *Relative timing*, donde un nodo establece un *offset* respecto a nodos vecinos.
- *Global synchronization*, donde todos los nodos se sincronizan con el reloj global.

Debido a que el último protocolo requiere de un GPS en cada nodo y la red es una red local, no se requiere de adquirir *hardware* extra para el algoritmo de sincronización de tiempo y se selecciona un método relativo. Entre los algoritmos descritos en la Sec. 6.10, el algoritmo seleccionado es ‘*Time synchronized mesh protocol*’ (TSMP, por sus siglas en ingles) [15]. Este protocolo funciona al conectar la capa física con la parte lógica del flujo de los datos. La capa física, al estar dividida en varios canales en la frecuencia de 2.4 GHz y la parte lógica, que requiere de múltiples saltos para que el mensaje llegue del emisor al receptor encuentran una conexión al utilizar un protocolo de sincronización de tiempo. Esta conexión se basa en dividir espacios de tiempo entre mensajes y que cada envío esta en uno de los múltiples espacios de tiempo y en un canal disponible determinado por una supertrama enviada por un nodo en común.

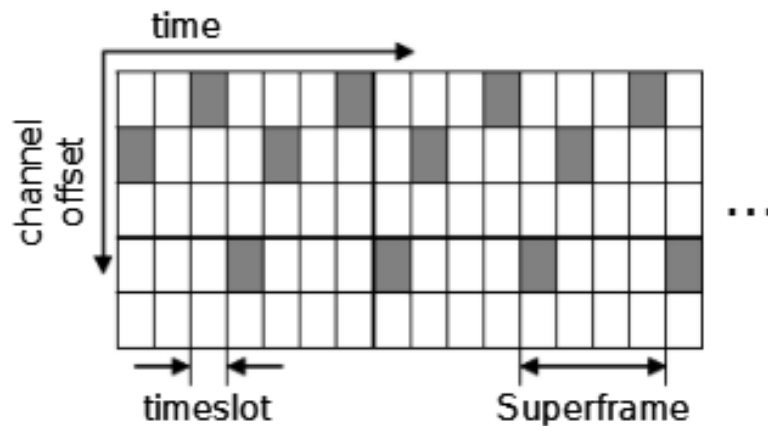


Figura 19: Matriz para una red con espacios de tiempo y 5 canales disponibles [15].

El algoritmo se desarrolló con el concepto de los espacios de tiempo. El coordinador es el nodo encargado de mandar el mensaje por medio de un *broadcast* a todos los miembros de la red en todos los canales disponibles para que activen la bandera de sincronización e inicien el temporizador y puedan enviar sus mensajes cuando el tiempo llegue al desfase particular respecto el momento en que se recibe el mensaje del coordinador como se observa

en la Figura 20. Una vez enviado el mensaje, se baja la bandera, se reinicia el temporizador y queda a la espera nuevamente del mensaje del coordinador para iniciar el temporizador nuevamente. Este algoritmo otorga robustez a la red ya que garantiza que cada miembro de la red envíe su mensaje en un tiempo determinado y se eviten las colisiones.

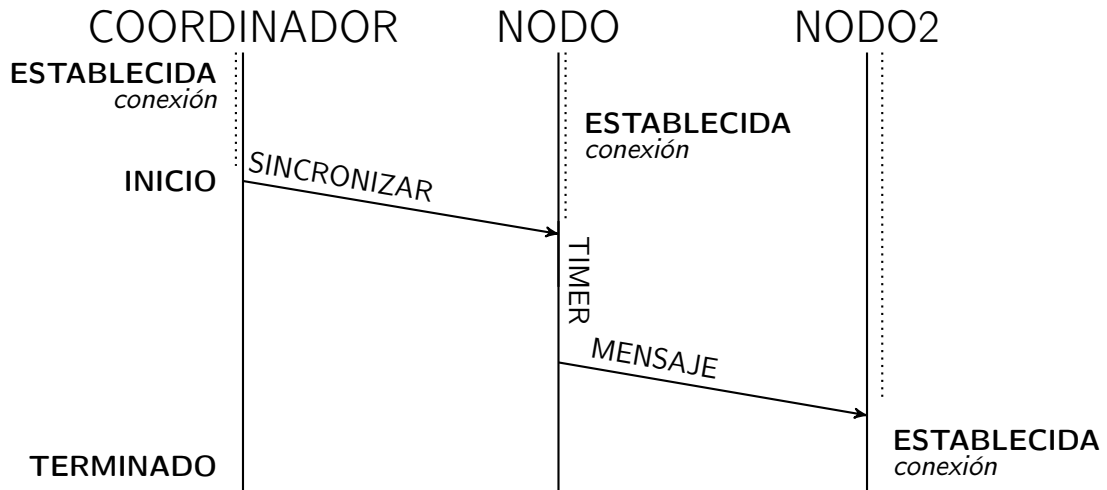


Figura 20: Secuencia de comunicación para el servicio de sincronización.

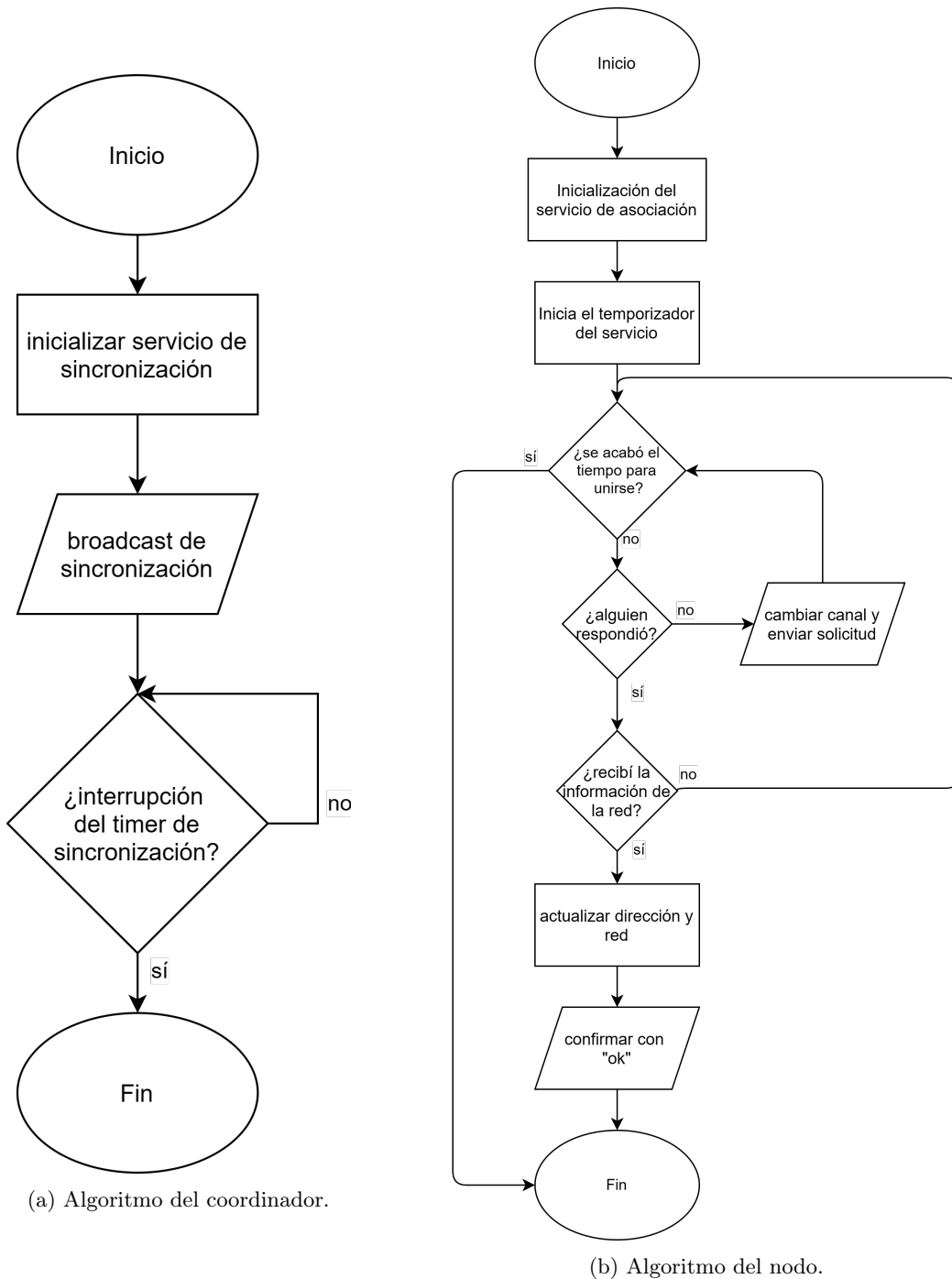


Figura 21: Algoritmos del servicio de sincronización.

### 8.3.3. Algoritmo de nodos conectados

Este servicio es necesario en la red para mantener actualizada la información de la red ya que dentro de las aplicaciones de enjambre se utilizan múltiples robots y si alguno llegara a fallar es necesario que la red se adapte a los cambios que sufre. Debido a este problema se implementó un algoritmo que es ejecutado por el coordinador de la red para recibir la información de los nodos y actualizar la información de la red.

En la Figura 23, el coordinador envía un mensaje a los nodos uno por uno esperando recibir un ACK como respuesta en un tiempo definido. En el caso que ocurre la interrupción del temporizador y el nodo nunca respondió con el ACK, el coordinador llega a la conclusión que el nodo está desconectado y actualiza la red. Cuando el coordinador termina de revisar todos los nodos conectados procede a enviar un paquete por medio de *broadcast* para que todos actualicen la información de la red.

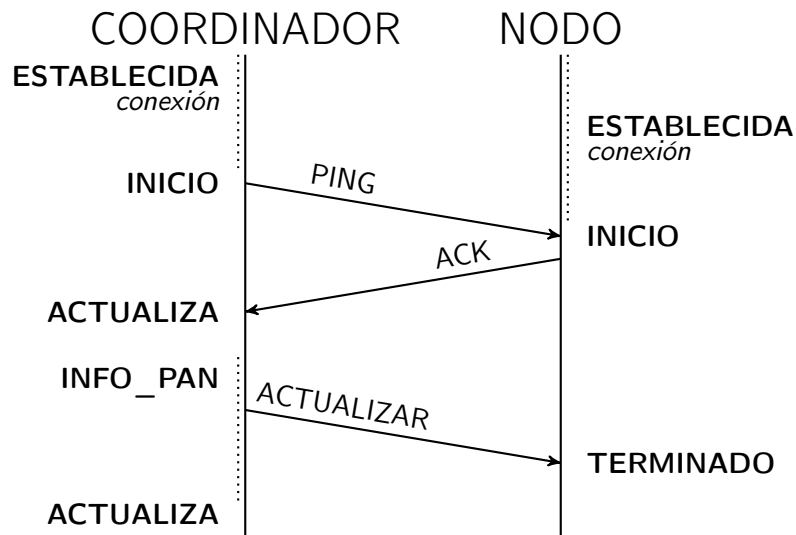
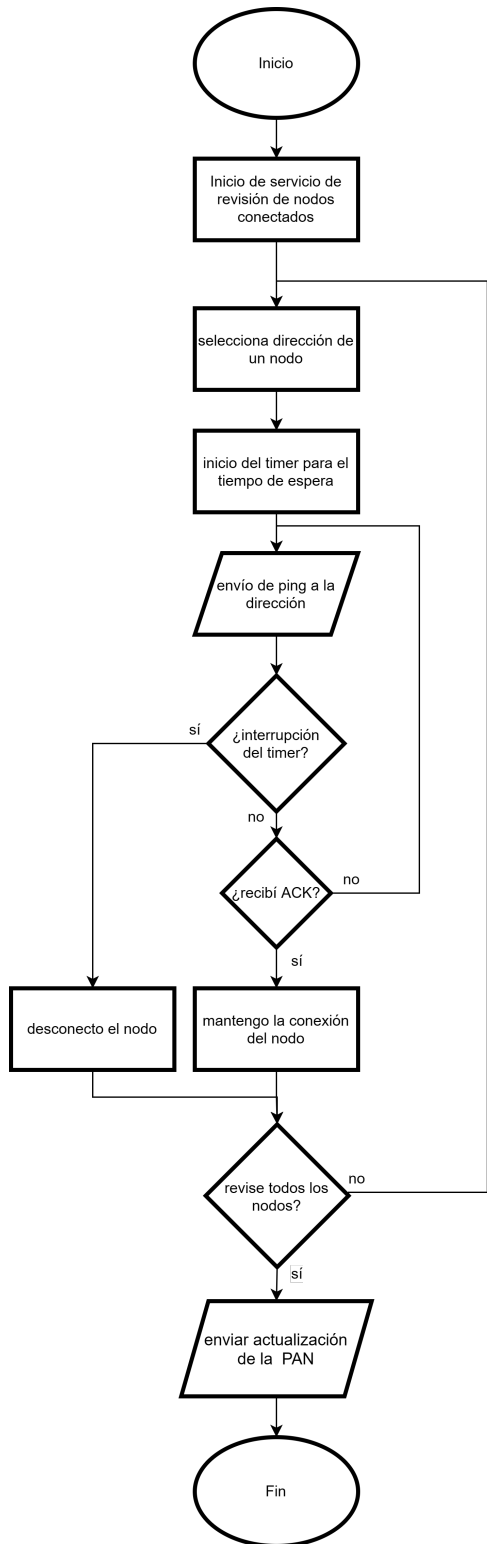
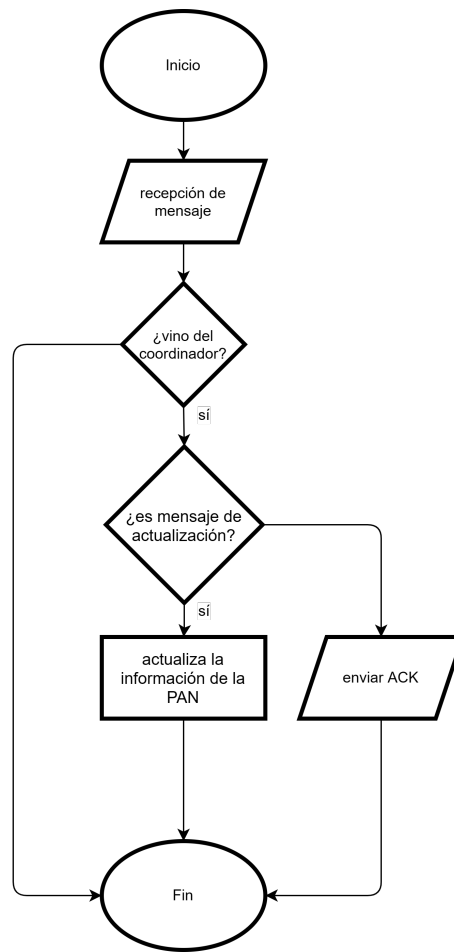


Figura 22: Secuencia de comunicación para el algoritmo de nodos conectados para una red con un nodo



(a) Algoritmo del coordinador.



(b) Algoritmo del nodo

Figura 23: Algoritmos de nodos conectados.

### 8.3.4. Servicio de elección de coordinador

Este servicio es necesario para que la robótica de enjambre mantenga su robustez ya que por medio de este servicio se elimina el problema de único punto de fallo descrito en la sección 6.11 y en la sección 7.2.3 cuando se mencionaba la importancia del aspecto de fiabilidad para la topología seleccionada. Por lo tanto, la capacidad de seleccionar otro coordinador cuando el actual falle resuelve el problema en una topología descentralizada.

Los algoritmos de bravucón y de anillo son estrategias básicas para resolver este problema. Sin embargo, se tomó un enfoque basado en el algoritmo *Raft*. Se utilizó este enfoque debido a la capacidad que tiene el transceptor MRF24J40MA de enviar un mensaje por *broadcast* y que el algoritmo enseña el mecanismo para mantener a los nodos o sistemas conectados a la red.

En el servicio de elección de coordinador se incluye un algoritmo para el coordinador y un algoritmo para los nodos. El algoritmo para el coordinador es trivial ya que únicamente debe de enviar un PING a todos los nodos en un tiempo  $T$ . Mientras que el algoritmo para los nodos es un poco más extenso.

Cada nodo espera recibir un PING del coordinador para estar al tanto que el coordinador sigue conectado. Este mensaje debe de llegar antes de un tiempo  $2T$ . Cuando transcurre el tiempo  $2T$  y no recibió ningún PING, el nodo inicia el servicio de elección de coordinador. El nodo cambia a candidato, activa un temporizador para caducar el servicio y envía un mensaje a toda la red para informar que el coordinador se desconectó y que ahora él es un candidato. Cada nodo recibe el mensaje y están enterados de la situación, pero no responden. Cuando transcurre el tiempo  $2T$  para cada uno de los nodos, revisan si alguien ya se postuló como candidato y confirman que se desconectó el coordinador enviando su voto al candidato que comenzó el servicio. Además, activan sus temporizadores esperando una confirmación del nuevo coordinador.

En el caso que existan dos nodos que se postularon como candidatos, ambos enviaron sus solicitudes a la red. Sin embargo, cada uno de los nodos votará únicamente por uno; por el que envió la solicitud de primero. Cuando el candidato recibe suficientes votos para confirmar su elección, cambia su configuración de candidato a coordinador y envía un *broadcast* a toda la red informando que hay un nuevo coordinador.

Si ocurre la interrupción del temporizador del candidato y no llegó a los suficientes votos, el candidato cambia su estado a un nodo más y espera el siguiente PING del coordinador. Si ocurre la interrupción del temporizador para los nodos que votaron y el candidato no tomó el puesto del coordinador, los nodos esperan el siguiente PING del coordinador anterior.

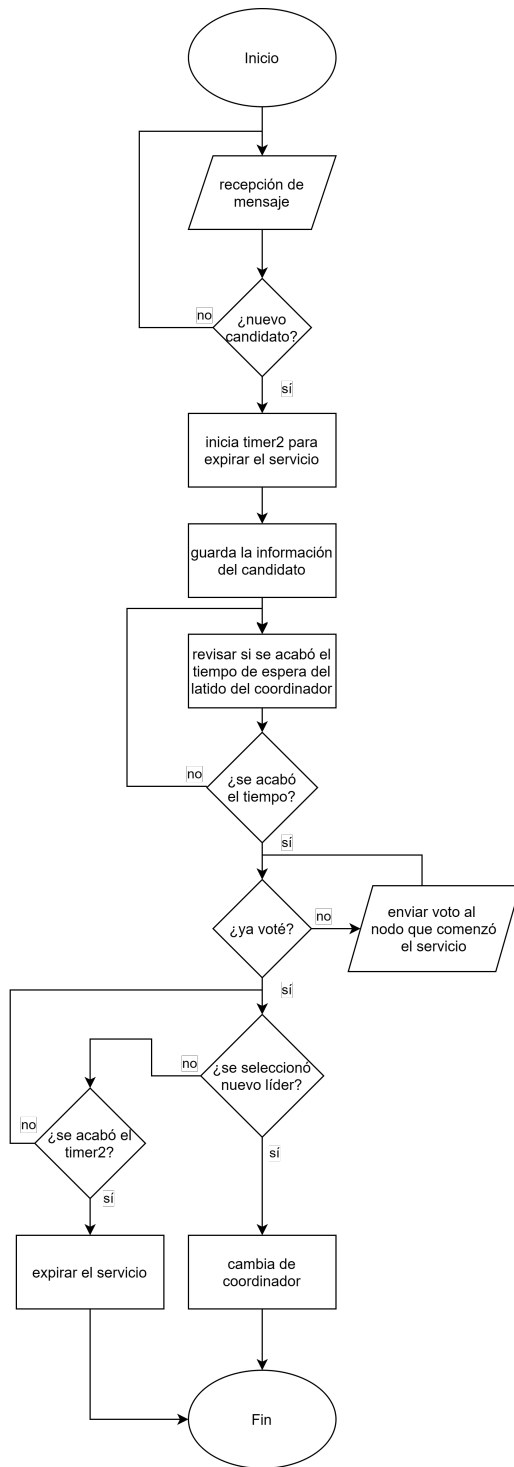


Figura 24: Algoritmo del nodo.

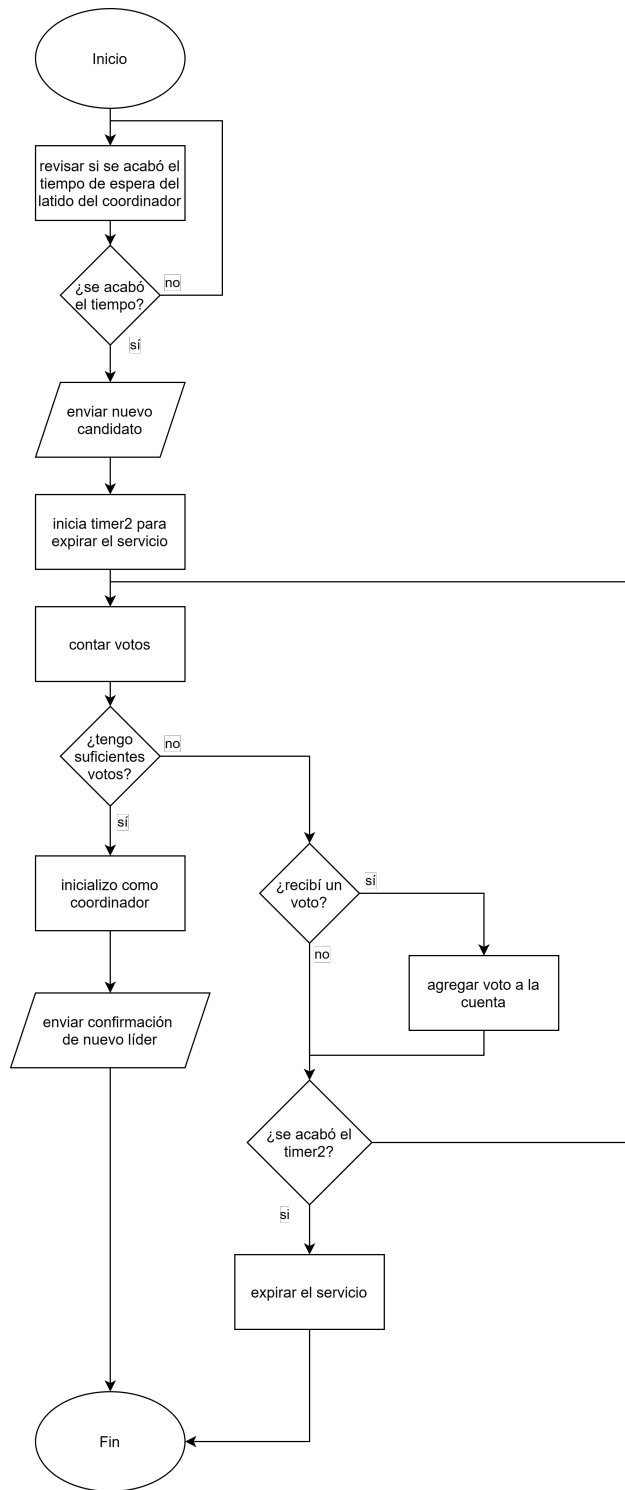


Figura 25: Algoritmo del candidato.



---

### Medición de la red

---

En la red que se construyó entorno a ZigBee es necesario que se determinen métricas que describan la red física. Como se ha presentado este trabajo, la transición de una simulación a la realidad trae problemas inherentes a la topología y tecnología que se seleccionaron. Por lo tanto, las métricas que se determinaron muestran el desempeño de la red construida y servirá para futuras investigaciones en la línea de robótica de enjambre para que se tenga un contexto real de la capacidad de la comunicación del enjambre.

En la tesis de Kui Liu [23], se encuentra una metodología para evaluar ZigBee y tener una referencia real de su desempeño. En el trabajo se encuentran cinco métricas estudiadas:

- Tasa de rendimiento efectiva
- Tamaño de la red
- Seguridad
- Consumo energético
- Alcance y retardo en paquetes

Sin embargo, para la medición de la red propuesta se enfocarán únicamente en la tasa de rendimiento efectiva, consumo energético y alcance y retardo de paquetes. No se enfocó en la seguridad y el tamaño de la red ya que tanto [23] y el presente trabajo tienen estas métricas definidas por el protocolo ZigBee.

ZigBee ofrece ocho niveles de seguridad para asegurar la integridad y confidencialidad de los datos. También, el tamaño de la red para Zigbee dicta como límite un poco más de sesenta y cinco mil nodos en la misma red.

Para la evaluación de las otras métricas se tuvieron estas metodologías.

## 9.1. Metodología para la tasa de rendimiento efectiva

Acorde a la documentación de Zigbee [33], la tasa de transferencia es de 250 kbps para la frecuencia de 2.4 GHz. Sin embargo, debido a los encabezados que se agregan en cada capa la tasa de la carga útil disminuye. Es por ello, que el experimento para evaluar la tasa de rendimiento efectiva consiste en determinar cuántos mensajes de la carga útil llegan desde el emisor al receptor en un segundo.

El emisor envía inmediatamente un mensaje cuando recibe una confirmación del receptor, que ocurre cuando el mensaje llega completo. Cuando el receptor recibe un paquete del emisor, este revisa si el tamaño y el contenido son los correctos o no. Si no es correcto, el paquete es descartado y no se toma en cuenta en el cálculo de la tasa efectiva.

Se construyó un programa para el emisor y otro para el receptor. El emisor envía una carga útil para los ciento dieciséis *bytes*. Los demás *bytes* de la trama en Zigbee son utilizados para los encabezados. Este paquete lo recibe el receptor y compara la carga útil para confirmar que el paquete llegó completo. Este proceso se observa en los diagramas de flujo de las Figuras 26 y 27 para el emisor y receptor, respectivamente.

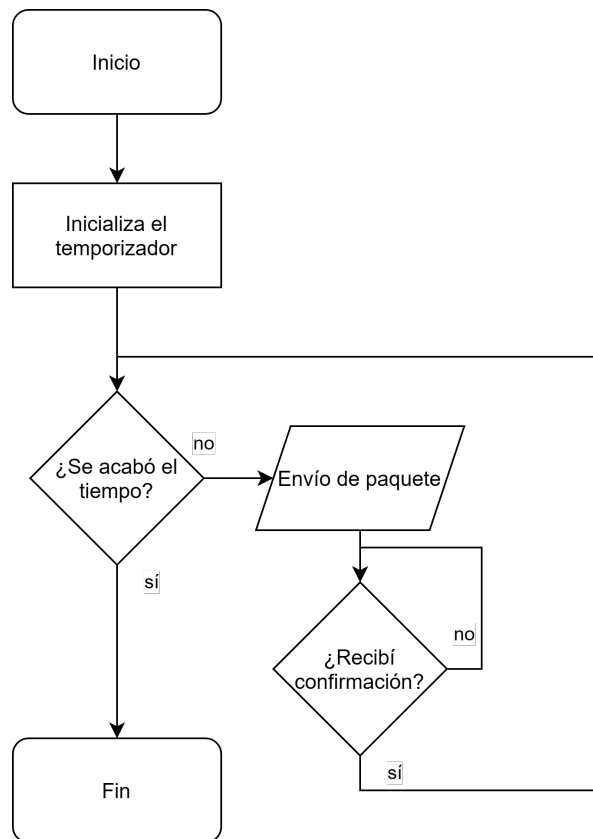


Figura 26: Algoritmo del emisor para la tasa de rendimiento efectiva.

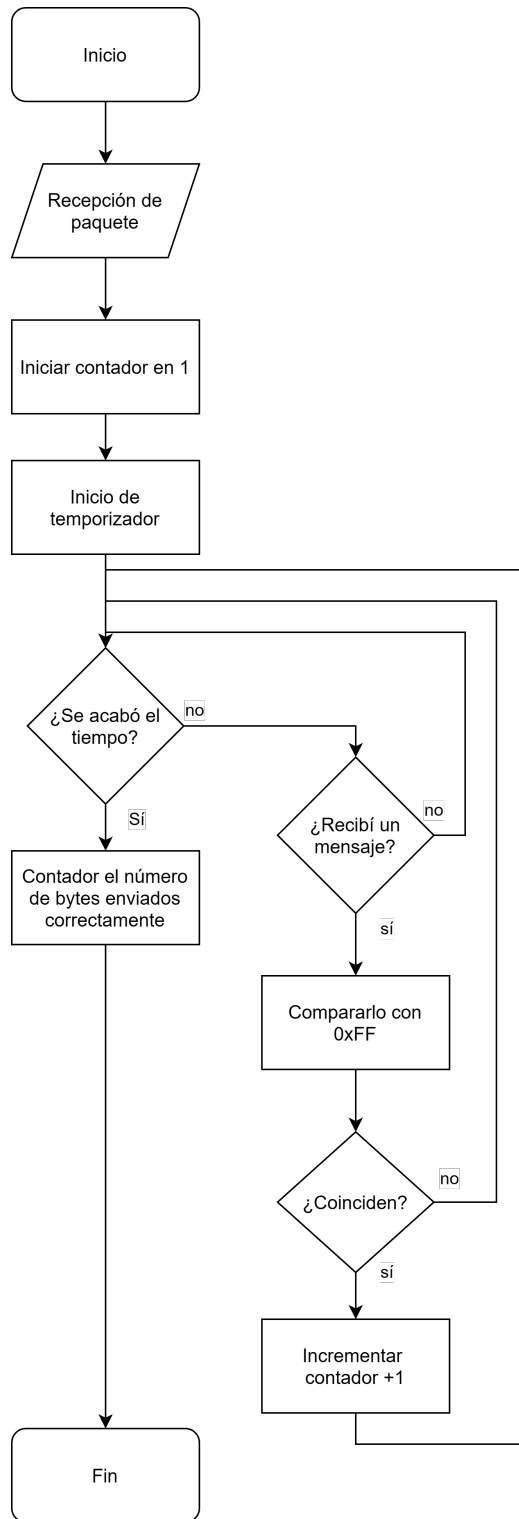


Figura 27: Algoritmo del receptor para la tasa de rendimiento efectiva.

## 9.2. Metodología para el consumo energético

El consumo energético es importante para las aplicaciones de robótica de enjambre debido a que los robots no estarán estáticos todo el tiempo. Por lo tanto, necesitaran una fuente de energía que pueden cargar como una batería 18650 o bien cables suficientemente largos para los robots, pero esta solución no es eficiente.

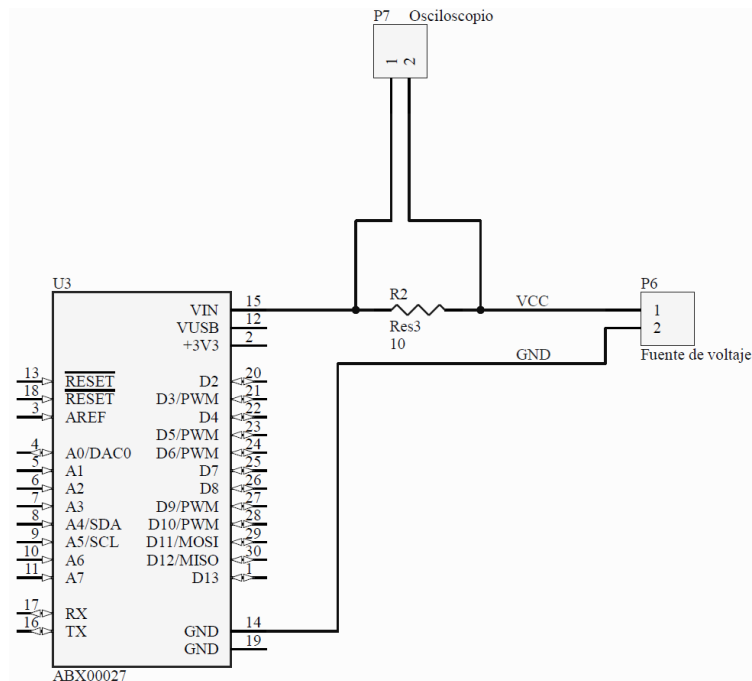


Figura 28: Circuito para medir la corriente consumida.

Entonces la medición del consumo energético se basa en determinar la corriente demandada por el microcontrolador para comunicarse con el transeptor y enviar y recibir paquetes. Se implementó el circuito de la Figura 28 en el cual se mide el diferencial de potencial en la resistencia de 10  $\Omega$  y mediante la ecuación:

$$V = I * R \quad (1)$$

Se despeja para la corriente (I) y se calcula con el diferencial de potencial medido y la resistencia de 10  $\Omega$ .

## 9.3. Metodología para el alcance y retardo en paquetes

Finalmente, se empleó una prueba similar a [23] para el alcance y retardo de paquetes siendo la única diferencia que se empleó la antena del PCB y no se sustituyó como en la tesis de Kui Liu.

Se calculó el tiempo de ida y vuelta, RTT, para determinar el retardo de paquetes. Además, las medidas se realizaron a distintas distancias y se contaron los paquetes exitosos y los fallidos para establecer la tasa de paquetes perdidos. De esta manera, se puede establecer una tolerancia máxima de paquetes perdidos para definir la distancia máxima a la que pueden estar los robots.

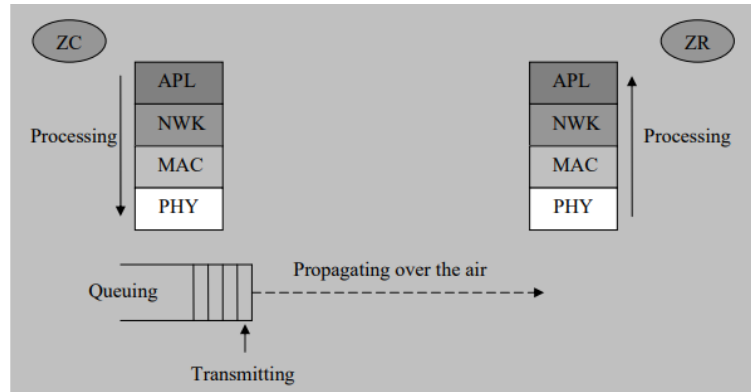


Figura 29: Proceso de transmisión de paquete [23].

Para el cálculo de RTT se consideran los siguientes tiempos según la Figura 29:

- Tiempo de procesamiento de la capa superior a la inferior en el emisor y de la inferior a la superior en el receptor.
- El tiempo de cola para poder enviar el mensaje. También se toma en cuenta el tiempo que le toma al nodo ganar el acceso al medio.
- Tiempo de transmisión a una velocidad de 250 kbps.
- Tiempo de propagación en el aire.

El proceso consiste en el envío de un mensaje y que el receptor lo procese para leerlo y comparar si es el mensaje correcto. Una vez confirmado el mensaje, se responde con una palabra de confirmación para terminar una vuelta. El tiempo de ida y vuelta comienza desde que se envió el primer paquete hasta que se recibe la palabra de confirmación. En el caso que se pierda el paquete en la transmisión, existe un temporizador que al llegar a cero se reenvía el mensaje; el tiempo de la retransmisión también se considera en el tiempo de ida y vuelta y el paquete que nunca llegó se considera como un paquete perdido. Se realizaron dos mil corridas para calcular el tiempo de ida y vuelta a distintas distancias.



### 10.1. Tasa de rendimiento efectiva

ZigBee teóricamente permite una transmisión de hasta 250kbps en la banda de 2.4 GHz. Sin embargo, en la realidad, la tasa de rendimiento efectiva disminuye, es decir, la tasa de rendimiento que contiene un mensaje (116 *bytes*) excluyendo los encabezados que se agregan por el protocolo no es la óptima.

En la prueba de tasa de rendimiento efectiva, se construyó un código que se implementó en dos nodos para medir la velocidad real. En esta prueba, el emisor transmite inmediatamente después que el receptor recibe una “S” o una “N” para confirmar la recepción y si el mensaje llegó completo. Por cada “S” recibida se sumaba al número de paquetes recibidos y por cada “N” recibida no se sumaba ya que el paquete no era el correcto.

El código que se construyó determina el número de *bytes* enviados en un segundo para poder determinar la velocidad de transferencia, kbps. El código se ejecutó dos mil veces a una distancia de un metro y medio para tener las velocidades experimentales que se tendrían en la mesa de pruebas de la UVG que tiene las medidas de 1.2\*0.9 m. En la siguiente Figura se muestra el porcentaje para las velocidades obtenidas en la prueba.

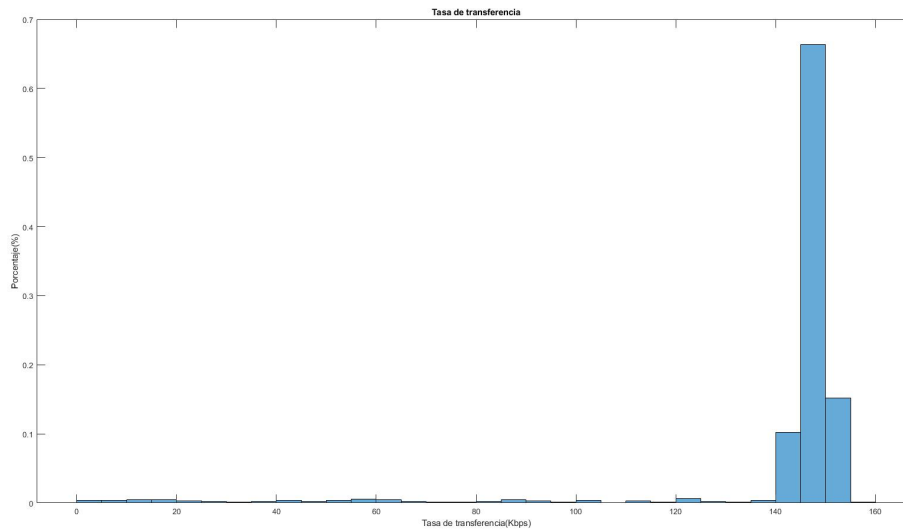


Figura 30: Histograma para la tasa de transferencia.

Como se observa en la Figura 30 la velocidad con mayor porcentaje está entre 140 y 160 kbps que se encuentra muy por debajo de los 250 kbps de la teoría. Adicionalmente, esta velocidad equivale al 91.75 % con una dispersión baja. La tasa de transferencia efectiva promedio es igual a 141.20 kbps que para aplicaciones de robótica de enjambre es suficiente.

## 10.2. Alcance y retardo de paquetes

Los escenarios para las pruebas fueron delimitados a espacios cerrados como salones y expuesto a distintas fuentes de interferencia como *bluetooth* o *Wifi*. Las distancias que se utilizaron fueron 2, 6 y 10 metros para abarcar las dimensiones de un salón cerrado. En las siguientes Figuras se observa que conforme se presentaban escenarios a una mayor distancia, el tiempo de ida y vuelta (RTT, por sus siglas en ingles) era mayor.

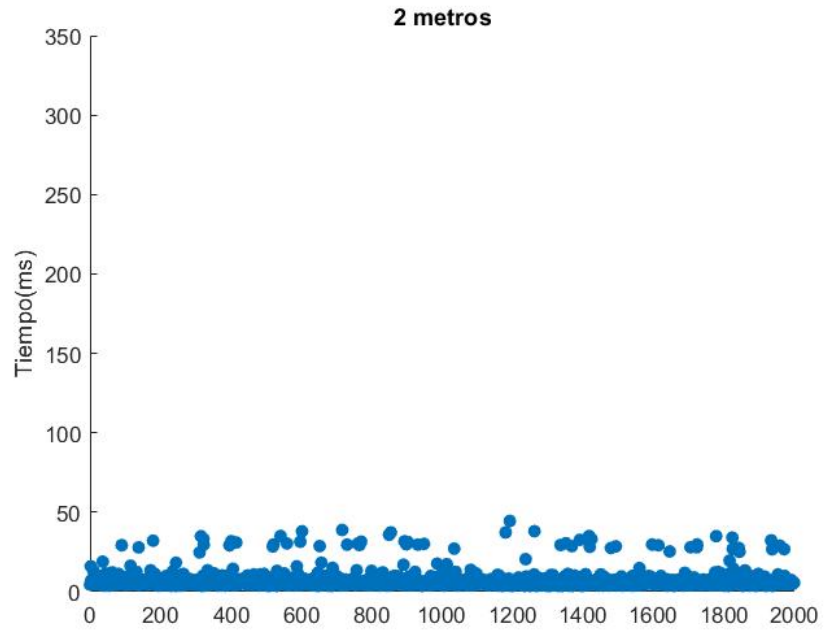


Figura 31: Diagrama de dispersión con una distancia de 2 metros.

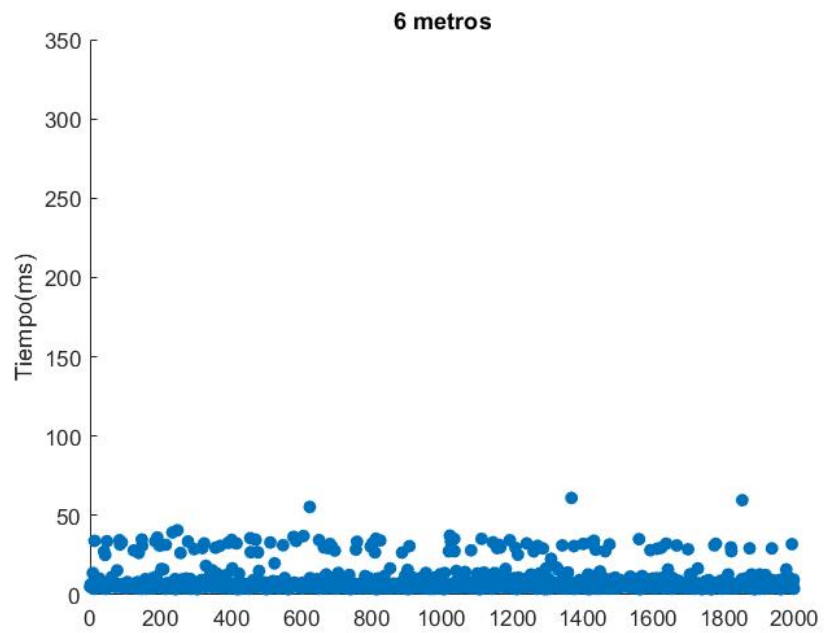


Figura 32: Diagrama de dispersión con una distancia de 6 metros.

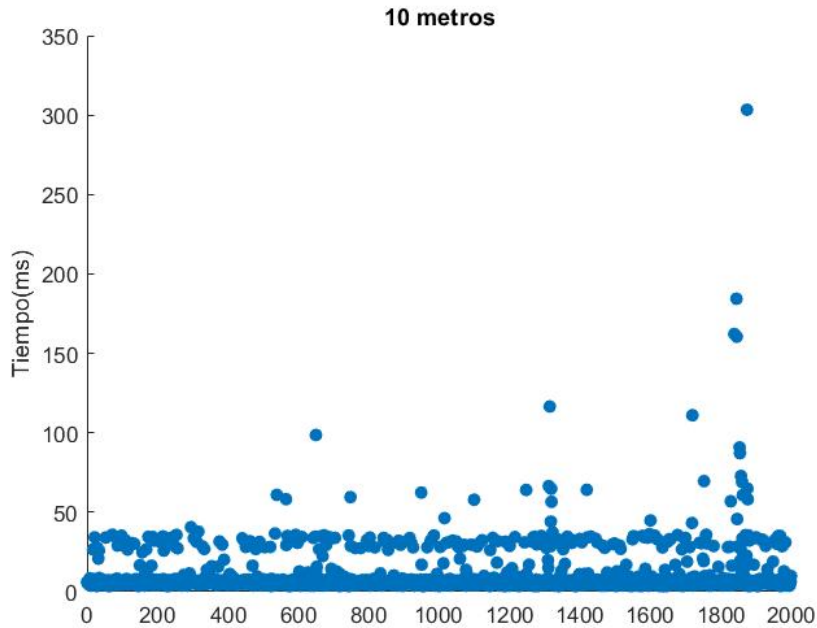


Figura 33: Diagrama de dispersión con una distancia de 10 metros.

Adicionalmente, se tiene una mayor pérdida de paquetes conforme la distancia incrementa como se muestra en el siguiente cuadro. Cabe mencionar que para condiciones de escenarios distintos: espacios abiertos, interferencia controlada, distinta antena, entre otros., el desempeño puede mejorar o empeorar.

Distancia (m)	RTT promedio (ms)	Pérdida de paquete (%)
2	6.8067	2.85
6	7.5057	5.05
10	9.7466	13.50

Cuadro 5: Comparación de distancias en un espacio cerrado.

### 10.3. Consumo energético

Se construyó el circuito de la Figura 28 para la medición de corriente en la transmisión y la recepción del transeptor. El circuito consiste en una fuente de voltaje, una resistencia de  $10 \Omega$  y el *shield* para el arduino nano. Para la medición se utiliza un osciloscopio que se conecta entre las terminales de la resistencia. Las mediciones de los voltajes para la transmisión y recepción son los siguientes.

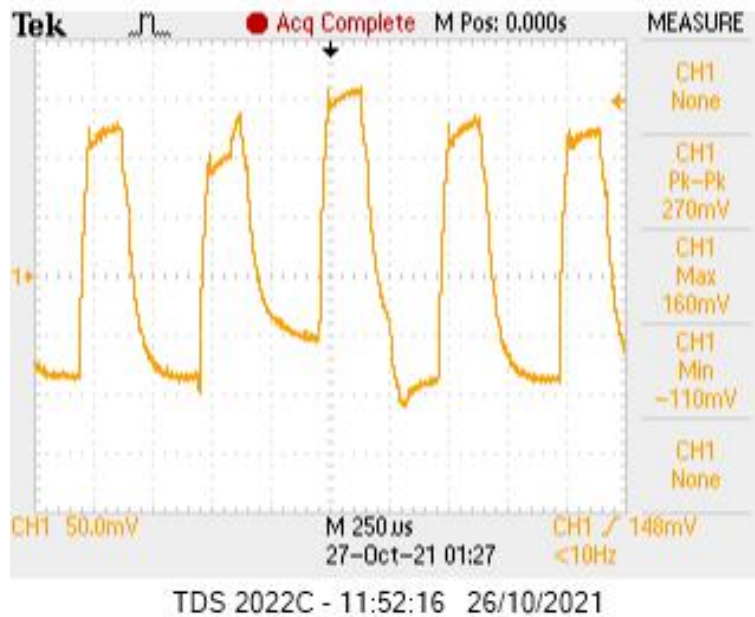


Figura 34: Medición del voltaje para la transmisión con el transceptor MRF24J40MA.

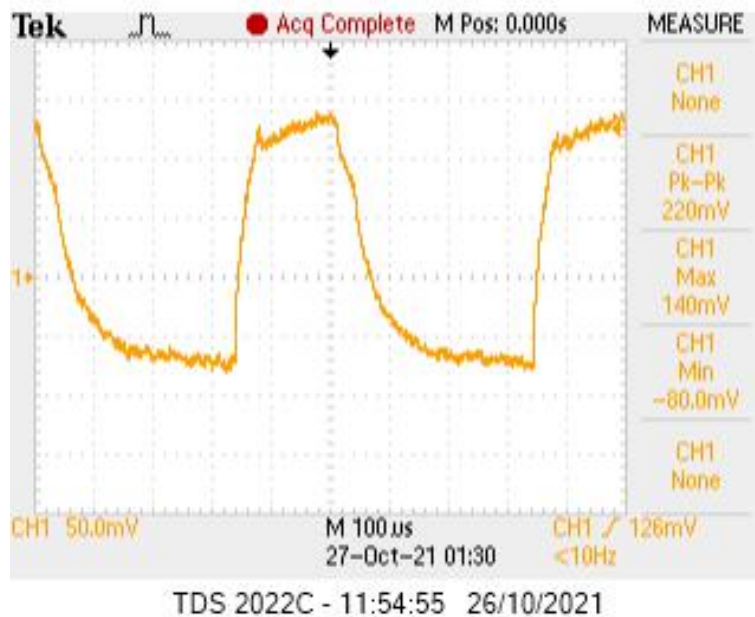


Figura 35: Medición del voltaje para la recepción con el transceptor MRF24J40MA.

Las mediciones en el osciloscopio fueron realizadas con un acople AC para analizar únicamente el cambio en el voltaje causado por el transceptor. Utilizando la ecuación 1 se despeja

para la corriente y se calcula utilizando la resistencia de  $10 \Omega$  y el voltaje pico a pico. El consumo de corriente para la transmisión es 27 mA y el consumo de corriente para la recepción es 22 mA para la recepción. La corriente consumida por el transceptor es consistente con la teoría de Zigbee que indica un consumo de 29 mA como se observa en el Cuadro 2. Ya que el transceptor es de bajo consumo tiene la capacidad de ser utilizado en aplicaciones con baterías como fuente de energía.

#### 10.4. Efectividad de los servicios

Se midió la efectividad de los servicios desarrollados: asociación, sincronización, nodos conectados y elección de coordinador. Para cada uno de los servicios se desarrolló un código que se ejecutó el servicio repetidamente por dos mil iteraciones a una distancia de 2, 6 y 10 metros para abarcar las dimensiones de un salón cerrado. En las siguientes tablas se muestran las iteraciones con éxito, es decir, que el servicio se ejecutó correctamente y el porcentaje que representan estas iteraciones con éxito respecto las dos mil ejecutadas para las distintas distancias.

<b>Distancia [m]</b>	<b>Iteraciones con éxito</b>	<b>Porcentaje de éxito</b>
2	1999	99.95 %
6	1981	99.05 %
10	1964	98.20 %

Cuadro 6: Comparación de éxitos para el servicio de asociación.

En el Cuadro 6 se muestra el resultado de asociar un nodo a la red creada por un coordinador. Se clasifica como éxito cuando el nodo después de modificar su dirección y su PAN manda el mensaje de confirmación al coordinador y éste actualiza la información de los nodos conectados.

<b>Distancia [m]</b>	<b>Iteraciones con éxito</b>	<b>Porcentaje de éxito</b>
2	2000	100.00 %
6	1972	98.60 %
10	1959	97.95 %

Cuadro 7: Comparación de éxitos para el servicio de sincronización.

En el Cuadro 7 se muestra el resultado de sincronizar el mensaje de cada nodo asociado a la red. La red con la que se realizaron las pruebas contiene un coordinador y dos nodos. Se clasifica como éxito cuando los mensajes de los nodos llegan en el orden correcto y dentro del tiempo esperado. Si llega un mensaje del segundo nodo conectado primero, el servicio falló. Si los mensajes no llegan en el tiempo esperado para dos nodos conectados, el servicio falló.

<b>Distancia [m]</b>	<b>Iteraciones con éxito</b>	<b>Porcentaje de éxito</b>
2	1999	99.95 %
6	1996	99.80 %
10	1998	99.90 %

Cuadro 8: Comparación de éxitos para el algoritmo que revisa los nodos conectados.

En el Cuadro 8 se muestra el resultado del proceso del coordinador al revisar si ningún nodo de la red se ha desconectado y actualizar la información de la red en todos los nodos. La red con la que se realizaron las pruebas contiene un coordinador y dos nodos. Se clasifica como éxito cuando todos los nodos responden al coordinador para informarle que siguen conectados. Cuando un nodo no llega a responder en el tiempo fijado el coordinador lo interpreta como si el nodo se ha desconectado y un fallo al servicio.

<b>Iteraciones</b>	<b>Iteraciones con éxito</b>	<b>Porcentaje de éxito</b>
100	88	88.00 %

Cuadro 9: Resultado del servicio para seleccionar nuevo coordinador.

En el Cuadro 9 se muestra el resultado de hacer fallar al coordinador y que la red seleccione un nuevo coordinador cien veces. La red que se construyó consistió en un coordinador y cuatro nodos conectados. Cuando todos los nodos se conectaron a la red y se dejó correr un tiempo un código de ejemplo en la red se prosiguió a desconectar el coordinador. Los nodos al no recibir el latido del coordinador inicializan el servicio de nuevo coordinador. Un nodo comienza antes que todos y solicita una votación para auto seleccionarse como coordinador, los demás nodos votan y se lleva a cabo la transición al nuevo coordinador. Cuando un nodo no logra hacer la transición al nuevo coordinador comenzará a intentar iniciar el servicio de nuevo coordinador, pero como no existen nodos en la red previa no se llevará a cabo. Luego de tres intentos asume que ya hay una red la cual no está conectado, por lo que ejecuta el servicio de asociación para unirse a la nueva red.

Se toma como acierto cuando un nodo logra tomar el lugar de coordinador y todos los nodos hacen la transición al nuevo coordinador ya sea porque cada uno logró hacer la transición cuando el servicio se ejecutó o porque se asociaron luego de que no estaban asociados a ninguna red.



Se comparó la topología de estrella propuesta por la fase previa con algunas descentralizadas y se determinó que la topología de malla se adecúa mejor a las aplicaciones de robótica de enjambre por medio de un estudio de compensación y utilizando las características de robótica de enjambre como criterio de decisión.

Se comparó y se seleccionó el protocolo Zigbee para la topología de malla seleccionada con base de un estudio de compensación. El bajo costo, bajo consumo energético y optimización para la topología seleccionada y utilizando las características de robótica de enjambre como criterio de decisión posicionaron a Zigbee sobre los demás protocolos.

Se desarrollo y se implementó un sistema de comunicación que incorpora cuatro servicios desarrollados: asociación a una nueva red, sincronización de mensajes, revisión de los nodos conectados a la red y elección de nuevo coordinador, con el fin de que la red sea fiable en su funcionamiento y resuelva las fallas asociadas a las fallas de punto único y colisión de paquetes.

Se determinó la efectividad de los servicios desarrollados para el sistema de comunicación en un espacio de dimensiones de la mesa de pruebas de la UVG. El servicio de asociación a una nueva red tuvo un éxito de 99.95 % en dos mil iteraciones, el servicio de sincronización de mensajes tuvo un éxito de 100.00 % en dos mil iteraciones, el servicio de revisión de los nodos conectados a la red tuvo un éxito de 99.95 % en dos mil iteraciones y el servicio de elección de nuevo coordinador tuvo un éxito de 88.00 % en cien interacciones.

Se determinó que el rendimiento de la red para un espacio de dimensiones de la mesa de pruebas de la UVG es de 160 kbps para la tasa de transferencia, 6.8067 ms para el tiempo de mandar un paquete y recibirlo de vuelta (RTT) y 27 mA para el consumo de corriente en el peor de los casos.



Se presentan las siguientes recomendaciones:

- La implementación de todos los servicios desarrollados en conjunto para evitar al máximo la colisión de datos.
- Medir la energía en el canal en el lugar de aplicación para evitar la interferencia en la red para seleccionar el canal con una energía menor a 40 dB. El método que está en la librería para medir la energía es “lqi(void)”.
- Se recomienda inicializar de primero el coordinador y luego los nodos uno por uno para que el continuo “broadcast” de los nodos intentando conectarse a una red no ocasione una colisión de datos.
- Utilizar las técnicas de enrutamiento desarrollados en los últimos años para aprovechar al máximo la topología de malla.
- Utilizar el sistema de comunicación en los sistemas físicos de robótica de enjambre desarrollados en los últimos años.



- 
- [1] C. M. Cianci, X. Raemy, J. Pugh y A. Martinoli, *Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics*, 2007.
  - [2] A. T. Khan, X. Cao, S. Li y Z. Milosevic, *Blockchain Technology with Applications to Distributed Control and Cooperative Robotics: A Survey*, ene. de 2019. DOI: 10.5430/ijrc.v2n1p36.
  - [3] Wikipedia, *E-puck mobile robot*, [https://en.wikipedia.org/wiki/E-puck\\_mobile\\_robot](https://en.wikipedia.org/wiki/E-puck_mobile_robot), Accessed: 2021-06-05, 2019.
  - [4] M. J. C. Martínez, ed., *Diseñar e implementar una red de comunicación inalámbrica para la experimentación en robótica de enjambre*, Tesis de Licenciatura, Universidad del Valle de Guatemala, 2019.
  - [5] N. World, *Cómo entender (y recordar) el modelo de red de 7 capas*. <https://www.networkworld.es/networking/como-entender-y-recordar-el-modelo-de-red-de-7-capas>, Accessed: 2021-03-30, 2017.
  - [6] M. Dorigo, M. Birattari y M. Brambilla, *Swarm robotics*, revision #138643, 2014. DOI: 10.4249/scholarpedia.1463.
  - [7] Z. Xi, ed., *The comparison of decentralized and centralized structure of network communication in different application fields*, 2019 International Conference on Management Science and Industrial Economy, Tianjin, China: Nankai University, 2019.
  - [8] B. Zhang y S. Gao, eds., *The Study of ZigBee Technology's Application in Swarm Robotics System*, 2nd International Conference on Artificial Intelligence, Jiaozuo, China: Management Science y Electronic Commerce, 2018.
  - [9] J. P. Dignani, ed., *Análisis del protocolo Zigbee*, Tesis de maestría, La Plata, Argentina: Universidad Nacional de La Plata, 2011.
  - [10] R. Kölln y A. Zimmermann, *Transparent Coordinator Failure Recovery for ZigBee Networks*. Sep. de 2009. DOI: 10.1109/ETFA.2009.5347103.
  - [11] Microchip, *MRF24J40MA*, <https://www.microchip.com/wwwproducts/en/MRF24J40MA#additional-features>, Accessed: 2021-06-05, 2016.

- [12] P. J. Baker y J. T. Whalen, *Survey of Trade Study Methods for Practical Decision-Making*, [https://www.nasa.gov/sites/default/files/survey\\_of\\_trade\\_study\\_methods\\_-\\_baker.pdf](https://www.nasa.gov/sites/default/files/survey_of_trade_study_methods_-_baker.pdf), Accessed: 2021-12-15, 2010.
- [13] D. D. Geetha y N. Tabassum, *A survey on clock synchronization protocols in wireless sensor networks*, 2017. DOI: 10.1109/SmartTechCon.2017.8358424.
- [14] B. Kaur y K. Amandeep, *A survey of time synchrhonization protocols for wireless sensor networks*, sep. de 2013.
- [15] K. Pister y L. Doherty, *TSMP: Time synchronized mesh protocol*, ene. de 2008.
- [16] S. J. Bigelow, *Single point of failure (SPOF)*, <https://searchdatacenter.techtarget.com/definition/Single-point-of-failure-SPOF>, Accessed: 2021-09-17, 2020.
- [17] Q. Mamun, S. M. Masum, M. Abdur y R. Mustafa, *Modified bully algorithm for electing coordinator in distributed systems*, ene. de 2004.
- [18] R. J. Anthony, *Systems Programming*, 2016. DOI: 10.1016/B978-0-12-800729-7.00006-6.
- [19] ETHEREUM, *Consensus mechanisms*, <https://ethereum.org/en/developers/docs/consensus-mechanisms/>, Accessed: 2021-09-17, 2021.
- [20] D. Ongaro y J. Ousterhout, eds., *In Search of an Understandable Consensus Algorithm*, The Advanced computing systems association, Philadelphia, PA: Stanford University, 2014.
- [21] R. M. D'Souza, S. Ramanathan y D. T. Lang, *Measuring performance of ad hoc networks using timescales for information flow*, revision #138643, 2003. DOI: 10.1109/INFCOM.2003.1208991.
- [22] S. A. Jyothi, A. Singla, P. B. Godfrey y A. Kolla, *Measuring and Understanding Throughput of Network Topologies*, 2016. DOI: arXiv:1402.2531.
- [23] K. Liu, *Performance Evaluation of ZigBee Network for Embedded Electricity Meters*, Master's thesis, Estocolmo, Suecia, 2009.
- [24] CISCO, *Troubleshooting Ethernet Collisions*, <https://www.cisco.com/c/en/us/support/docs/interfaces-modules/port-adapters/12768-eth-collisions.html>, Accessed: 2021-09-14, 2006.
- [25] AVI Networks, *Single Point Of Failure*, <https://avinetworks.com/glossary/single-point-of-failure/>, Accessed: 2021-09-14, 2021.
- [26] A. H. Mousa, N. T. Mohammed y E. A. Mohammed, *EFCNT: An evaluation framework for computer's network topologies*, ago. de 2019. DOI: 10.1063/1.5123126.
- [27] I. F. Jassam y S. K. Guirguis, *Comparison of network topology types*, [https://gs.alexu.edu.eg/upload/Students/1404/1404703\\_2020-2021\\_Fall/1404-3-160\\_assignment\\_1.pdf](https://gs.alexu.edu.eg/upload/Students/1404/1404703_2020-2021_Fall/1404-3-160_assignment_1.pdf), Accessed: 2021-09-14, 2020.
- [28] J. Soparia y N. Bhatt, *A Survey on Comparative Study of Wireless Sensor Network Topologies*, ene. de 2014. DOI: 10.5120/15175-3255.
- [29] A. Shrestha y L. Xin, *A Performance Comparison of Different Topologies for Wireless Sensor Networks*, 2007. DOI: 10.1109/ths.2007.370059.
- [30] P.-S. Chong y K. C. Arun, *Performance evaluation of wireless and wired network topologies*, 2021.

- [31] DigiKey, *Módulos y módems de transceptores RF*, <https://www.digikey.com/>, Accessed: 2021-09-15, 2021.
- [32] ESPRESSIF, *ESP-NOW*, <https://www.espressif.com/en/products/software/esp-now/resources>, Accessed: 2021-09-15, 2016.
- [33] C. standart alliance, *Zigbee FAQ*, <https://zigbeealliance.org/zigbee-faq/>, Accessed: 2021-09-15, 2021.
- [34] O. Horyachyy, *Comparison of Wireless Communication Technologies used in a Smart Home: Anaysis of wireless sensor node based on Arduino in home automation scenario*, Thesis for the degree of Master, Karlskrona, Sweden, 2017.
- [35] ZWAVE, *Learn about Z-Wave*, <https://www.z-wave.com/learn>, Accessed: 2021-09-15, 2020.
- [36] J.-r. Lin, T. Talty y O. Tonguz, *On the Potential of Bluetooth Low Energy Technology in Vehicular Applications*, feb. de 2015. DOI: 10.1109/MCOM.2015.7010544.
- [37] ANT, *ANT BASICS*, [urlhttps://www.thisisant.com/developer/ant/ant-basics](https://www.thisisant.com/developer/ant/ant-basics), Accessed: 2021-09-15, 2021.
- [38] Adafruit, *XBee Module - ZB Series S2C 2mW with Wire Antenna XB24CZ7WIT-004*, <https://www.adafruit.com/product/968#description>, Accessed: 2021-06-05, 2016.
- [39] Arduino, *Arduino Nano*, <https://store.arduino.cc/arduino-nano>, Accessed: 2021-08-01, 2016.
- [40] D. Scazzoli, A. Kumar, N. Sharma, M. Magarini y G. Verticale, *A Novel Technique for ZigBee Coordinator Failure Recovery and Its Impact on Timing Synchronization*, sep. de 2016. DOI: 10.1109/PIMRC.2016.7794552.



## **14.1. Librería para el transceptor MRF24J40MA**

La librería desarrollada para el transceptor MRF24J40MA se encuentra en el siguiente repositorio: Redes de Comunicación

