

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN ALGORITMO PARA LA ASIGNACIÓN DE DIRECCIONES GLOBALES EN UNA TOPOLOGIA LINEAL UTILIZANDO EL ESTÁNDAR 802.15.4

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

KEVIN DARIO SIMBAÑA CUSICAGUA

kevin.simbana@epn.edu.ec

DIRECTOR: ING. CARLOS ROBERTO EGAS ACOSTA, Msc.

carlos.egas@epn.edu.ec

Quito, Diciembre 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Kevin Darío Simbaña Cusicagua, bajo mi supervisión.

A handwritten signature in blue ink, consisting of several loops and a long horizontal stroke at the bottom.

ING. CARLOS EGAS, Msc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo , Kevin Darío Simbaña Cusicagua, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



KEVIN DARIO SIMBAÑA CUSICAGUA

DEDICATORIA

A mi madre, que con su ejemplo me enseñó el valor del trabajo duro, ejemplo que hoy me ha ayudado a finalizar otra meta más.

Y a mi familia, que me brinda su apoyo en esta travesía universitaria.

AGRADECIMIENTO

Agradezco a mis padres por ser los pilares de mi vida y de mi formación académica. Sin ellos no tendría razón de existir.

Agradezco a mi familia por preocuparse siempre por mí y alentarme continuamente para finalizar mi carrera.

Agradezco a mis compañeros de universidad, por estar ahí y echarme una mano siempre cuando lo necesitaba.

Agradezco de manera especial al Ing. Carlos Egas por brindarme su tiempo, apoyo y guía en la realización de esta tesis.

Finalmente agradezco a la Escuela Politécnica Nacional por haberme brindado la oportunidad de desarrollarme como profesional en tan prestigiosa institución.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS.....	XI
ÍNDICE DE CÓDIGOS.....	XII
RESUMEN.....	XIII
ABSTRACT	XIV
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE	2
1.3 MARCO TEÓRICO	4
1.3.1 REDES INALÁMBRICAS DE SENSORES	4
1.3.2 ESTÁNDAR IEEE 802.15.4.....	7
1.3.3 ZigBee	24
1.3.4 6LowPAN	27
1.3.5 MÓDULO TRANSECTOR ZIGBIT DE ATMEL.....	28
1.3.6 HERRAMIENTA DE DESARROLLO DE ATMEL.....	32
2. METODOLOGÍA.....	35
2.1. DISEÑO DEL ALGORITMO.....	35
2.1.1. REQUERIMIENTOS DEL ALGORITMO.....	35
2.1.2. DESCRIPCIÓN GENERAL DEL ALGORITMO.....	35

2.2.	IMPLEMENTACIÓN DEL ALGORITMO	46
2.2.1.	ANÁLISIS DE REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DEL ALGORITMO.....	46
2.2.2.	IMPLEMENTACIÓN DEL CÓDIGO	48
2.3.	EVALUACIÓN DE LOS RETARDOS	64
3.	RESULTADOS Y DISCUSIÓN	69
3.1.	CONFORMACIÓN DE LA TOPOLOGIA LINEAL.....	69
3.1.1	COMPONENTES USADOS EN LAS PRUEBAS.	69
3.1.2	IMPLEMENTACIÓN DEL PROTOTIPO DE RED INALÁMBRICA DE PRUEBA.....	69
3.2.	PRUEBAS DE MEDICIÓN DE POTENCIA.....	71
3.3.	PRUEBAS REALIZADAS DEL ALGORITMO DE ASIGNACIÓN DE DIRECCIONES	72
3.3.1.	PRUEBA 1.....	73
3.3.2.	PRUEBA 2.....	76
3.3.3.	PRUEBA 3.....	80
3.3.4.	PRUEBA 4.....	84
3.3.5.	PRUEBA 5.....	87
3.3.6.	PRUEBA 6.....	91
3.4.	COMPARACIÓN DE TIEMPOS.....	95
3.4.1	COMPARACIÓN CON TIEMPO TEÓRICO.	95
3.4.2	COMPARACIÓN DE TIEMPOS CON OTROS PROTOCOLOS.....	95
3.5.	ANÁLISIS DE LOS RESULTADOS	97
4.	CONCLUSIONES Y RECOMENDACIONES	98
4.1.	CONCLUSIONES.....	98
4.2.	RECOMENDACIONES.....	99

5. REFERENCIAS BIBLIOGRÁFICAS.....	100
ANEXOS.....	104

ÍNDICE DE FIGURAS

Figura 1.1. Arquitectura de un nodo sensor	4
Figura 1.2 Componentes de una WSN.....	5
Figura 1.3. Topología en estrella	9
Figura 1.4. Topología punto a punto	9
Figura 1.5. Topología cluster-tree.	10
Figura 1.6. Arquitectura del estándar IEEE 802.15.4.....	11
Figura 1.7. Canales definidos por el estándar IEEE 802.15.4	13
Figura 1.8. Estructura del paquete de capa física.	14
Figura 1.9. Interfaz entre capa MAC y la capa superior	15
Figura 1.10. Estructura de la supertrama.	17
Figura 1.11. Transferencia de datos de dispositivo a coordinador con modo beacon habilitado	18
Figura 1.12. Transferencia de datos de dispositivo a coordinador con modo beacon no habilitado.	18
Figura 1.13. Transferencia de datos de coordinador a dispositivo con modo beacon habilitado.	19
Figura 1.14. Transferencia de datos de coordinador a dispositivo con modo beacon no habilitado.	20
Figura 1.15. Formato de la trama de datos	21
Figura 1.16. Formato de la trama ACK.....	21
Figura 1.17. Formato de la trama de comandos.....	22
Figura 1.18. Formato de la trama beacon	22
Figura 1.19. Información de direccionamiento dentro del encabezado de trama.	23
Figura 1.20. Protocolo ZigBee.....	24
Figura 1.21. Red inalámbrica de sensores <i>Long-Thin</i>	26

Figura 1.22. Arquitectura del protocolo TCP/IP y 6LowPan.....	27
Figura 1.23. Vista general de la placa de desarrollo ZigBit ATZB-256RFR2-XPRO.	30
Figura 1.24. Conexión de la interfaz JTAG para programar la placa	31
Figura 1.25. Conexión de la fuente de alimentación a la placa de desarrollo.	31
Figura 1.26. Arquitectura por bloques que utiliza Atme.	32
Figura 2.1. Configuraciones de red iniciales.....	36
Figura 2.2. Diagrama del funcionamiento normal del algoritmo.....	39
Figura 2.3. Diagrama de enlaces fallidos en la recepción de peticiones de asignación. ...	41
Figura 2.4. Diagrama en el que el nodo borde o frontera no recibe identificador.....	42
Figura 2.5. Diagrama en el que el nodo cercano o intermedio no recibe identificador.	43
Figura 2.6. Clasificación de la señal según niveles de referencia.....	44
Figura 2.7. Diagrama del funcionamiento en caso de nodo fallido.....	45
Figura 2.8. Herramienta programadora Atmel-ICE.	46
Figura 2.9. Sniffer Zigbee CC2531 USB Dongle	47
Figura 2.10. Estructura de archivos al crear un proyecto.	48
Figura 2.11. Diagrama de flujo para el proceso de asignación de direcciones (Parte 1)..	50
Figura 2.12. Diagrama de flujo para el proceso de asignación de direcciones (Parte 2)..	51
Figura 2.13. Tiempos que intervienen en la asignación de direcciones.....	65
Figura 2.14. Proyección del algoritmo de direccionamiento para cientos de nodos.....	68
Figura 3.1 Componentes del prototipo de prueba.	69
Figura 3.2. Fotografías de pruebas en un pasillo de la Escuela Politécnica Nacional.....	70
Figura 3.3. Medición de niveles de señal recibida según la distancia.....	72
Figura 3.4 Topología lineal de prueba.....	72
Figura 3.5. Captura de tramas prueba 1.....	74
Figura 3.6. Tramas en escenario de asignación normal.	75

Figura 3.7. Captura de tramas prueba 2.....	77
Figura 3.8. Tramas en escenario de enlaces fallidos prueba 2.....	78
Figura 3.9. Captura de tramas prueba 3.....	81
Figura 3.10 Tramas en escenario de enlaces fallidos prueba 3.....	82
Figura 3.11. Captura de tramas prueba 4.....	85
Figura 3.12. Tramas en escenario de enlaces fallidos prueba 4.....	86
Figura 3.13. Captura de tramas prueba 5.....	88
Figura 3.14. Tramas en escenario de enlaces fallidos prueba 4.....	89
Figura 3.15. Captura de tramas prueba 6.....	92
Figura 3.16. Tramas en escenario de enlaces fallidos prueba 4.....	93
Figura 3.17. Prototipo 1 de tesis de red 6LowPan	96

ÍNDICE DE TABLAS

Tabla 1.1. Características de la capa física del estándar IEEE 802.15.4	12
Tabla 3.1. Medición de Potencia	71
Tabla 3.2. Tiempo de ejecución del algoritmo en un escenario de asignación sin errores	75
Tabla 3.3. Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 2	79
Tabla 3.4 Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 3	82
Tabla 3.5. Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 4	86
Tabla 3.6. Tiempo de ejecución del algoritmo en un escenario de nodo fallido prueba 5.90	
Tabla 3.7. Tiempo de ejecución del algoritmo en un escenario de nodo fallido prueba 5.94	
Tabla 3.8. Tiempo obtenidos de una red 6LowPan [38].....	96
Tabla 3.9 Resumen de tiempos de asignación del algoritmo desarrollado.....	97

ÍNDICE DE CÓDIGOS

Código 2.1 Configuración de LEDs y botón SW2.....	52
Código 2.2 Configuración de parámetros de la red PAN.....	52
Código 2.3. Configuración de direcciones aleatorias iniciales.....	53
Código 2.4. Función principal main.....	54
Código 2.5 Definición de librerías y constantes.....	54
Código 2.6. Definición de Variables Globales	55
Código 2.7. Estructura que contiene los campos de la trama IEEE 802.15.4.....	55
Código 2.8. Estructura que contiene la información de direccionamiento.....	56
Código 2.9. Definición de objeto tipo Enum para distinguir los tipos de procesos.	56
Código 2.10. Envío de broadcast de asignación para sondeo de nodos	57
Código 2.11. Inicio del timer 1 después de enviar el broadcast.....	57
Código 2.12. Extracción, almacenamiento y contabilización de tramas de respuesta del nodo asignador.....	58
Código 2.13. Envío de respuesta a mensaje broadcast de asignación.....	59
Código 2.14. Ordenamiento de la tabla de direccionamiento según el nivel de potencia.....	59
Código 2.15. Ordenamiento según umbrales de potencia para caso de nodo fallido.	60
Código 2.16. Lógica de envío de los identificadores.	61
Código 2.17. Confirmación de envío de tramas de asignación para prevenir enlaces fallidos.	62
Código 2.18. Reenvío de identificador al nodo asignador siguiente.	62
Código 2.19. Recepción de tramas de asignación.	64
Código 2.20. Configuración de dirección.....	64
Código 3.1. Código de medición de potencia recibida. Código comentado	93

RESUMEN

El presente trabajo de titulación tiene como objetivo desarrollar un algoritmo de asignación de direcciones globales en una topología lineal con el estándar IEEE 802.15.4. En el primer capítulo se revisa las características generales de una red inalámbrica de sensores. Se revisa el estándar IEEE 802.15.4 así como los protocolos ZigBee y 6LoWPAN incluyendo sus métodos de direccionamiento. Se estudia el módulo transceptor ZigBit de Atmel y su herramienta de desarrollo, elementos necesarios para codificar el algoritmo.

En el segundo capítulo se muestra el diseño del algoritmo con su implementación en código. Además, se realiza una evaluación de los retardos de asignación proponiendo un modelo matemático que permite proyectar el tiempo que requiere el algoritmo en asignar direcciones en un escenario ideal.

En el tercer capítulo se realizan las pruebas correspondientes que permiten validar el correcto funcionamiento del algoritmo y se realizan las comparaciones pertinentes para evaluar los tiempos del algoritmo de asignación de direcciones frente a otros protocolos.

Por último, en el cuarto capítulo se muestran las conclusiones y recomendaciones obtenidas del presente proyecto.

PALABRAS CLAVE: IEEE 802.15.4, direccionamiento, algoritmo, retardos, tiempo, WSN, ZigBee, 6LoWPAN, Atmel.

ABSTRACT

The objective of this project is to develop an algorithm for addressing global addresses in a linear topology with the IEEE 802.15.4 standard. In the first chapter, the general characteristics of a wireless sensor network are reviewed. The IEEE 802.15.4 standard is reviewed, as well as the ZigBee and 6LoWPAN protocols, including their addressing methods. Atmel's ZigBit transceiver module and its development tool, elements necessary to code the algorithm, are studied.

The second chapter shows the design of the algorithm with its implementation in code. In addition, an evaluation of the assignment delays is carried out, proposing a mathematical model that allows projecting the time required by the algorithm to assign addresses in an ideal scenario.

In the third chapter, the corresponding tests are carried out that allow validating the correct functioning of the algorithm and the pertinent comparisons are made to evaluate the times of the addressing algorithm compared to other protocols.

Finally, the fourth chapter shows the conclusions and recommendations obtained from this project.

KEYWORDS: IEEE 802.15.4, addressing, algorithm, delays, time, WSN, ZigBee, 6LoWPAN, Atmel.

1. INTRODUCCIÓN

El uso de las redes inalámbricas de sensores cada día va en aumento, así como sus aplicaciones en la industria. Como consecuencia, ha aumentado la necesidad de buscar nuevas formas de optimizar los recursos usados por este tipo de redes representando así una ventaja económica en su operación y mantenimiento.

Los nodos inalámbricos usados en este tipo de redes presentan limitaciones en su capacidad de cálculo y en su tiempo de operación, a causa de su funcionamiento con baterías. Lo que ocasiona que el reto de las investigaciones en esta área sea mejorar la vida útil de las baterías minimizando los efectos negativos que producen los retardos en el consumo de energía.

El direccionamiento es otro aspecto importante en la operación normal de una red inalámbrica de sensores. En la actualidad la responsabilidad de asignar direcciones que permitan tener una conectividad de extremo a extremo está dada a niveles de red con protocolos tales como ZigBee o 6LoWPAN. Los métodos de direccionamiento de estos protocolos requieren de un procesamiento elevado y un alto control de la red ocasionando limitaciones en redes lineales multisalto a gran escala. Estos inconvenientes también han incrementado la necesidad de implementar soluciones óptimas para sistemas de redes inalámbricas de sensores que trabajan con una topología de red lineal.

El trabajo de titulación busca encontrar un algoritmo que permita optimizar la asignación automática de direcciones globales con mínimos tiempos de retardo traducido en un mínimo consumo de energía para poner operativa una red lineal multisalto a gran escala.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

Desarrollar un algoritmo para la asignación de direcciones globales en una topología lineal utilizando el estándar 802.15.4.

Los objetivos específicos del Proyecto Técnico son:

- Analizar de manera general el estándar IEEE 802.15.4.
- Crear un algoritmo para la asignación de direcciones globales.
- Implementar el prototipo para evaluar el algoritmo.
- Evaluar la propuesta respecto a otras tecnologías como 6LoWPAN y ZigBee.

1.2 ALCANCE

En la propuesta se va a desarrollar un algoritmo que asigne automáticamente direcciones globales a nodos que conforman topologías lineales multisalto a gran escala. Las direcciones de enlace que proporciona el estándar IEEE 802.15.4 [6] serán utilizadas como direcciones globales que permitan tener una conectividad de extremo a extremo sin necesidad de asignar direcciones de red al nodo, utilizando información únicamente del nivel de capa de enlace. El algoritmo propuesto además dará soluciones cuando existan nodos averiados y enlaces fallidos durante el proceso de asignación de direcciones, lo cual asegurará confiabilidad en el proceso.

La asignación de direcciones será de forma secuencial, con una numeración ascendente iniciando con la dirección cero en el nodo frontera de la topología lineal. Para determinar la dirección a asignar se realizará un proceso de detección de nivel de potencia que permitirá detectar los nodos que están dentro de la zona de cobertura de un nodo asignador de direcciones y en función de la potencia recibida asignar las direcciones de manera secuencial de acuerdo a si los nodos no asignadores están más cercanos o alejados del nodo asignador de direcciones. El nodo más alejado de la zona de cobertura se convertirá en un nuevo nodo asignador de direcciones y de esta manera continuará con el proceso de asignación de direcciones. Dentro de la zona de cobertura de un nodo asignador de direcciones se colocarán al menos dos nodos a la derecha y dos nodos a la izquierda.

Los nodos operarán en el modo no ranurado definido en el estándar 802.15.4 [6] que utiliza el protocolo CSMA/CA. El algoritmo requiere que durante el proceso de asignación de direcciones se envíen en la trama 802.15.4 nuevos comandos o mensajes, los cuales serán definidos durante el proceso de desarrollo del algoritmo.

Para evaluar el algoritmo desarrollado, se implementará un prototipo de red de al menos seis nodos sensores inalámbricos ubicados en una topología lineal multisalto, se utilizarán módulos Atmel-Microchip que son capaces de trabajar directamente con el estándar IEEE 802.15.4, específicamente se usarán los modelos ATZB-25 [7] en donde su principal componente es el microcontrolador de 8 bits ATMEGA-256RFR2[8]. Los nodos utilizados serán programados usando el software de Atmel Studio 7 y el lenguaje de programación C/C++.

Para evaluar los beneficios del algoritmo desarrollado se analizarán y se medirán los tiempos requeridos para asignar las direcciones a los nodos en el prototipo de red. Estos tiempos obtenidos se compararán con los tiempos obtenidos mediante pruebas utilizando

protocolos como 6LoWPAN, adicionalmente en el presente trabajo se pretende buscar librerías que permitan al protocolo ZigBee operar en redes multisalto y realizar una comparación adicional que permita validar las ventajas de esta propuesta.

Para los escenarios de pruebas y para constatar que el algoritmo asigna correctamente las direcciones de forma secuencial se utilizarán al menos 6 nodos con los cuales se probará el algoritmo desarrollado, considerando nodos y enlaces fallidos. En el caso de 6LoWPAN y ZigBee las pruebas de asignación de direcciones se realizarán en base a implementaciones ya existentes que funcionen en condiciones similares (no necesariamente iguales) al prototipo de red usado para probar el algoritmo desarrollado.

Para obtener los tiempos de asignación de direcciones de los prototipos implementados con los nodos ATZB-25, 6LoWPAN y ZigBee se utilizará un sniffer de red que permitirá observar la correcta asignación de direcciones en cada uno de los dispositivos usados. Como las pruebas se realizarán con un número limitado de nodos, se desarrollará una expresión matemática que permita proyectar el valor del retardo para el caso de tener cientos de nodos. Los tiempos de asignación de direcciones serán medidos considerando nodos dañados y enlaces fallidos

Al finalizar este trabajo de titulación el algoritmo propuesto será implementado en un prototipo de red que constará de al menos 6 nodos, el algoritmo será implementado en cada uno de los nodos para probar su funcionamiento. Para evaluar el funcionamiento correcto del algoritmo en el prototipo se lo evaluará en tres escenarios. En el primer escenario se realizarán pruebas en la ausencia de ruido. En el segundo escenario se considerará nodos averiados, para lo cual determinados nodos se apagarán intencionalmente al inicio del proceso de asignación de direcciones. En el tercer escenario se considerará realizar pruebas con enlaces fallidos, para lo cual mediante programación se simulará la no recepción de tramas consecuencia de un enlace fallido. Las pruebas de funcionamiento a realizar en los tres escenarios involucran la utilización de un sniffer para así poder observar y comprobar que las direcciones han sido asignadas de forma secuencial en la red lineal multisalto. Además, se medirá el tiempo que demora el algoritmo en asignar direcciones a los nodos.

1.3 MARCO TEÓRICO

1.3.1 REDES INALÁMBRICAS DE SENSORES

1.3.1.1 Definición de redes inalámbricas de sensores.

Una red inalámbrica de sensores (del inglés Wireless Sensor Network, WSN), es una red inalámbrica que consiste en un grupo de pequeños nodos inalámbricos que se encuentran distribuidos espacialmente y trabajan cooperativamente para actuar como generadores de datos, así como retransmisores de información en redes de comunicación. [1] [2]

1.3.1.2 Componentes de una WSN

Una red inalámbrica de sensores consta de los siguientes componentes [3][4]:

- **Sensores:** son dispositivos que capturan estímulos físicos y producen una respuesta medible ante un cambio de estado, como la luminosidad o temperatura, tienen el objetivo de transformar la señal capturada en señal eléctrica para ser interpretada por un controlador.
- **Nodos Inalámbricos:** también llamados motas, del inglés “mote”, por su tamaño reducido. Son dispositivos electrónicos que poseen la capacidad de capturar, procesar y transmitir inalámbricamente la información proveniente del medio en el que se encuentren ubicados. La arquitectura de un nodo inalámbrico se indica en la Figura 1.1



Figura 1.1. Arquitectura de un nodo sensor [4].

- **Gateway:** elemento de red cuyo objetivo es interconectar la red inalámbrica de sensores con una red de datos externa, actúa como puente de interconexión entre dos diferentes tipos de redes.

- **Estación Base:** posee la tarea de recolectar datos provenientes de los nodos para el adecuado tratamiento de la información.
- **Red Inalámbrica:** red basada por lo general en el estándar IEEE 802.15.4 u otros estándares como ZigBee, 6LoWPAN, WirelessHART, etc.

La Figura 1.2 muestra los componentes que conforman una WSN.

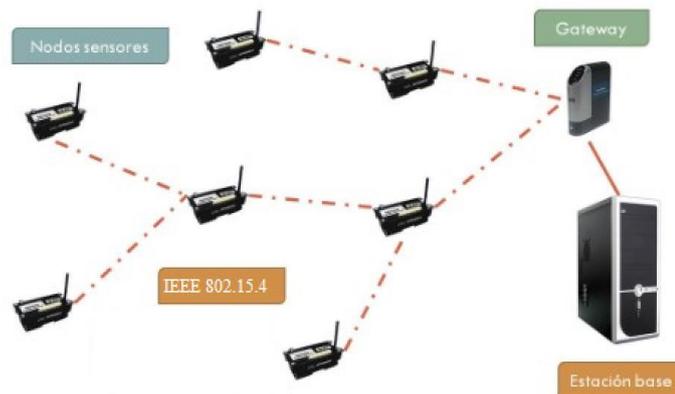


Figura 1.2 Componentes de una WSN [5]

1.3.1.3 Características de una WSN

Una WSN usualmente consiste en una gran cantidad de nodos de bajo costo, baja potencia y con capacidad de cálculo limitada, que generalmente se comunican sobre un medio inalámbrico a una distancia limitada, con el objetivo de cumplir una tarea o aplicación en común, como por ejemplo el monitoreo de carreteras, oleoductos, zonas fronterizas, etc.

Comparado con redes inalámbricas tradicionales, una red inalámbrica de sensores tiene las siguientes limitantes y características únicas [6]:

- **Gran despliegue de nodos:** La cantidad de nodos usados en un campo de interés específico, es usualmente densa. El número de sensores en la red puede variar en un orden de cientos o miles de nodos.
- **Uso de baterías:** Las baterías son usualmente la fuente de alimentación de los nodos sensores. En la mayoría de las situaciones, los nodos son colocados en ambientes desfavorables donde es demasiado difícil o incluso imposible cargar o reemplazar las baterías.

- Limitación en uso de energía, procesamiento y almacenamiento: Los nodos sensores son altamente limitados en capacidad de procesamiento, almacenamiento y energía.
- Auto configuración: Los nodos sensores son usualmente desplegados aleatoriamente sin un plan previo o ingeniería. Una vez desplegados, los nodos tienen que configurarse autónomamente a sí mismos para conformar una red de comunicación.
- Aplicaciones determinadas: Una red de sensores es normalmente diseñada y desplegadas para cumplir una aplicación específica. Los requerimientos de diseño de la red cambian con su tipo de aplicación.
- Baja confiabilidad: Los nodos sensores son usualmente desplegados en ambientes hostiles y operan sin cuidado, por lo que tienden a sufrir daños físicos o fallos.
- Redundancia de datos: En la mayoría de las aplicaciones de una red de sensores, se usa una gran cantidad de nodos, los cuales colaboran para cumplir una tarea de censado en común. Debido a esto, los datos detectados por múltiples nodos sensores suelen tener redundancia en la red de sensores.

1.3.1.4 Clasificación WSN

De acuerdo con el número de saltos entre un nodo sensor y la estación base, una red de sensores puede clasificarse en dos tipos de redes: redes monosalto y redes multisalto.

Las redes monosalto son redes en las cuales todos los nodos sensores se conectan directamente a la estación base y, como su nombre lo indica, la información se transmite directamente desde el nodo sensor hasta la estación base.

Las redes multisalto son redes en las cuales la información emitida por los nodos sensores es retransmitida a la estación base mediante uno o más nodos intermedios.

En general, una red monosalto tiene una arquitectura de red más simple y, por lo tanto, es más fácil de controlar. Es adecuada en áreas de detección pequeñas y para aplicaciones con nodos sensores escasamente desplegados. Las redes multisalto poseen una gama más amplia de aplicaciones a un costo de mayor complejidad en el control de la red. [4][6]

1.3.2 ESTÁNDAR IEEE 802.15.4.

El estándar IEEE 802.15.4 forma parte de un grupo de especificaciones desarrolladas por la IEEE para operar en redes inalámbricas de área personal (Wireless Personal Area Networks, WPAN). El objetivo de este estándar es especificar los niveles físicos y de acceso al medio (MAC) en redes inalámbricas. Además, de definir estos dos niveles, el estándar también apunta a lograr un cierto nivel de coexistencia con otros dispositivos inalámbricos.

Las especificaciones de diseño del estándar logran proveer a las redes inalámbricas de sensores: un bajo costo, en aplicaciones que requieren una distancia de comunicación corta; un bajo consumo de energía, para maximizar la vida útil de las baterías; una baja tasa de transmisión, asumiendo que la cantidad de datos transmitida es pequeña y poco frecuente; baja complejidad y alta flexibilidad, para ser soportadas por dispositivos inalámbricos con poca capacidad de procesamiento.[7]

Las principales características del estándar son las siguientes [8]:

- Tasas de transmisión de información de hasta 20 kbps a 868 MHz, 40 kbps a 915 Mhz y 250 kbps a 2.4 Ghz.
- Dos modos de direccionamiento: direccionamiento corto con 16 bits y direccionamiento extendido con 64bits.
- Compatible con dispositivos de baja latencia.
- Método de acceso al canal mediante CSMA/CA¹ y CSMA/CA Ranurado.
- Gestión de energía para garantizar el consumo de baja potencia.
- Puede operar en topología estrella o topología punto a punto.
- Transmisión fiable mediante el uso de acuses de recibo.
- Detección de energía.
- Indicador de calidad del enlace LQI².(*Link Quality Indicator*, Indicador de calidad del enlace)

¹ CSMA/CA, por sus siglas en inglés "*Carrier Sense Multiple Access with Collision Avoidance*" es un protocolo de control de acceso a redes que permite que múltiples estaciones utilicen un mismo medio de transmisión

² LQI, por sus siglas en inglés "*Link Quality Indicator*" es una indicación de la calidad de los paquetes de datos recibidos por el receptor.

1.3.2.1 Dispositivos

El estándar IEEE 802.15.4 define dos tipos de dispositivos [8]:

- **Dispositivos de funcionalidad completa (FFD – *Full Function Device*):** Los dispositivos FFD son capaces de actuar en distintos modos:
 - **Coordinador de la red de área personal (PAN):** es el dispositivo principal de la red, puede iniciar la comunicación, terminarla y enrutarla a través de la red.
 - **Coordinador:** dispositivo con la funcionalidad de sincronización con otros dispositivos cuando se emplea *beacons*³, el coordinador necesita estar asociado con un coordinador PAN ya que no puede crear su propia red.
 - **Dispositivo ordinario:** dispositivo que opera como un nodo final.
- **Dispositivos de funcionalidad reducida (RFD – *Reduced Function Device*):** Los dispositivos RFD fueron definidos para permitir la implementación de dispositivos extremadamente simples que requieren mínimos recursos en términos de procesamiento y capacidad de almacenamiento, características que tienen un impacto directo en la implementación de dispositivos de bajo costo. Los dispositivos RFD pueden comunicarse solo con un dispositivo FFD a la vez y no tienen la capacidad de ser coordinadores de una red PAN.

1.3.2.2 Topologías IEEE 802.15.4

Una red inalámbrica que opere con el estándar IEEE 802.15.4 puede tener dos topologías básicas: topología estrella y topología punto a punto. Otro tipo de topología que puede ser soportada por el estándar, pero no es parte de su especificación, es la topología tipo árbol o denominada también *cluster tree*. [12]

- **Topología estrella:** está formada alrededor de un dispositivo con funcionalidad completa llamado coordinador de la red PAN⁴, el coordinador actúa como un concentrador que centraliza la comunicación con los dispositivos finales que pueden ser dispositivos FFD o RFD. El coordinador de la red PAN es el único dispositivo en la red PAN que forma un enlace directo con los demás dispositivos presentes en la red. Ver Figura 1.3.

³ *Beacons* son tramas que poseen información acerca de la red inalámbrica y se transmiten periódicamente.

⁴ PAN, por sus siglas en inglés "*Personal Area Network*" conecta dispositivos electrónicos dentro del área inmediata de un usuario. El tamaño de una PAN oscila entre unos centímetros y unos metros.

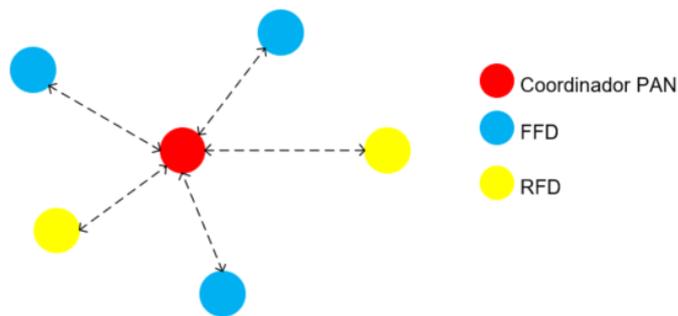


Figura 1.3. Topología en estrella [12].

- **Topología punto a punto:** este tipo de topología permite la comunicación punto a punto sin la participación directa de un coordinador de red PAN, cada dispositivo con funcionalidad completa o FFD tiene la capacidad de formar múltiples enlaces directos con otros dispositivos presentes en la red. Los dispositivos con funcionalidad reducida o RFD también pueden formar parte de la topología, pero solo como dispositivos finales o periféricos, dado su falta de capacidad para retransmitir información. Ver Figura 1.4.

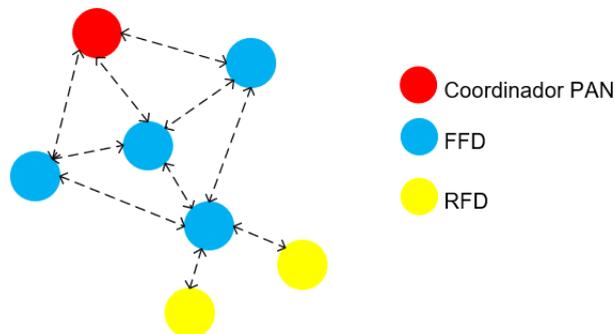


Figura 1.4. Topología punto a punto [12]

- **Topología tipo árbol:** está formada de manera jerárquica. La red inicia desde un nodo FFD denominado raíz, continuando de manera sucesiva con nodos denominados hijos que pueden ser dispositivos FFD o RFD. En una red tipo árbol solo un dispositivo asume el rol de coordinador de red PAN. Alrededor de cada nodo coordinador FFD se puede agrupar otra red tipo árbol con una relación padre - hijo, dando origen a una red *multicluster*. Ver Figura 1.5.

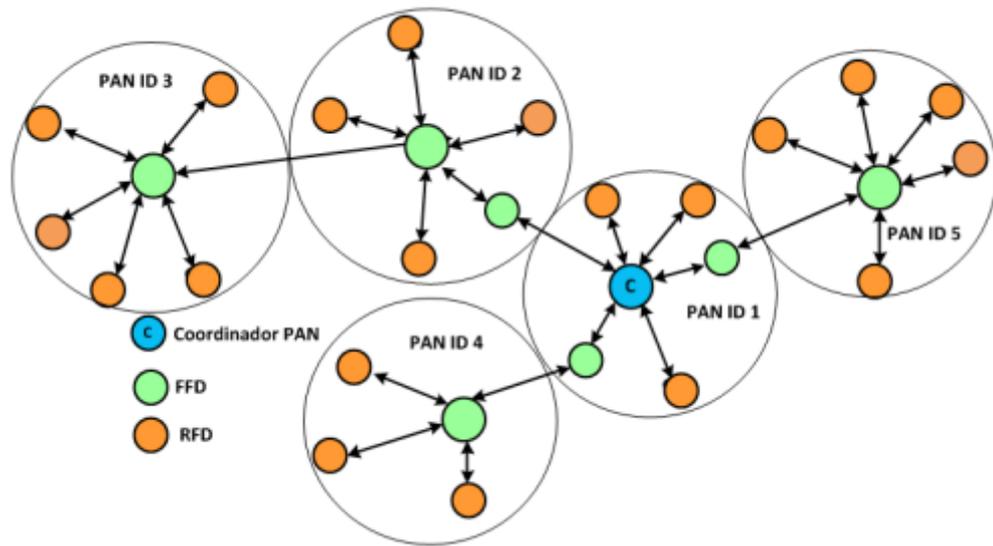


Figura 1.5. Topología *cluster-tree* [15].

- **Topología lineal:** una topología lineal se puede considerar como un caso especial de una topología tipo árbol en la cual solo se tiene una rama o ruta para toda la red. La topología lineal aprovecha las ventajas en términos de cobertura y alcance de una red *multicluster*, dando la posibilidad a una comunicación multisalto.

1.3.2.3 Arquitectura del estándar

La especificación IEEE 802.15.4 se orienta a cubrir la parte inferior del modelo OSI⁵ (*Open System Interconnection*) y en coexistir con otras especificaciones y dispositivos inalámbricos. Las capas inferiores que define el estándar son: la capa física y la capa de acceso al medio. La figura 1.3 muestra las capas que conforman la arquitectura del estándar IEEE 802.15.4.

⁵ OSI, por sus siglas en inglés "*Open Systems Interconnection*" describe una estructura con siete capas para las actividades de red.

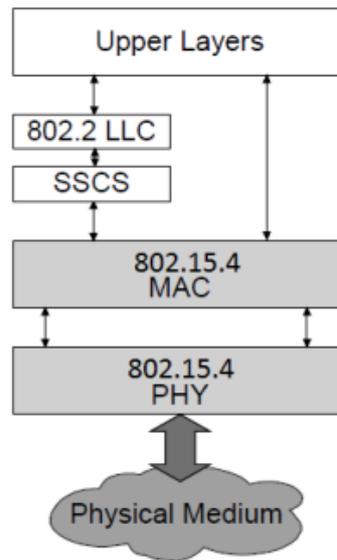


Figura 1.6. Arquitectura del estándar IEEE 802.15.4 [8].

1.3.2.3.1 Capa Física

La capa física del estándar IEEE 802.15.4 provee la interfaz con el medio físico en donde la comunicación real ocurre.

La capa física tiene las siguientes funcionalidades [8]:

- Activación y desactivación del transceptor de radio.
- Detección de energía.
- Obtención de la calidad del enlace.
- Selección de los canales definidos.
- Transmisión y recepción de datos.
- Verificar si el canal tiene libre acceso.

Además, la capa es responsable por el establecimiento del enlace de radiofrecuencia entre dos dispositivos, la modulación y demodulación de bits, y la sincronización entre el transmisor y el receptor. [9]

El estándar IEEE 802.15.4 opera en tres bandas de frecuencia no licenciadas, que se diferencian según el rango de frecuencia de operación, las velocidades de transmisión y

la modulación. El estándar IEEE 802.15.4 especifica un total de 27 canales *half-duplex*⁶ compartidos para las tres bandas de radiofrecuencia, cuya división entre canales y la diferencia entre las bandas de frecuencia definidas por la capa física se presentan en la Tabla 1.1.

Tabla 1.1. Características de la capa física del estándar IEEE 802.15.4 [8]

Bandas de Frecuencia	Rangos de frecuencia (MHz)	Modulación	Número de canales	Tasa de bits	Zona de utilización
868 MHz	868-868.6	BPSK	0 (1 canal)	20 Kb/s	Europa
915 MHz	902 – 928	BPSK	1-10 (10 canales)	40 Kb/s	Norte América
2400 MHz	2400 - 2483.5	O-QPSK	11-26 (16 canales)	250 Kb/s	Resto del mundo

El espectro de frecuencias de la banda de 868 MHz soporta un solo canal en el rango de frecuencias de 868 y 868.6 MHz, la banda de 915 MHz soporta diez canales que están separados 2MHz distribuidos entre el rango de 902 MHz y 928 MHz, la banda de 2.4 GHz soporta 16 canales que empiezan desde los 2400 MHz hasta los 2483.5 MHz con una separación de 5 MHz entre cada canal. La distribución de los canales definida por el estándar se puede apreciar en la figura 1.4.

⁶ *Half-duplex*: ambos dispositivos pueden transmitir y recibir datos en los medios, pero no pueden hacerlo en forma simultánea.

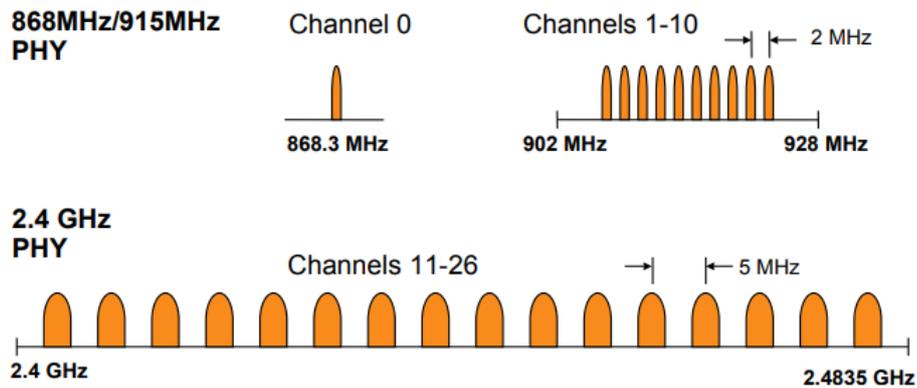


Figura 1.7. Canales definidos por el estándar IEEE 802.15.4 [10].

➤ *Detección del nivel de energía*

El estándar incorpora la funcionalidad de detección de energía (ED, *Energy Detection*) que puede ser usada por capas superiores para evitar interferencia entre las distintas comunicaciones de radio.

La detección de energía consiste en estimar el nivel de la señal recibida dentro del ancho de banda definido por un canal del estándar IEEE 802.15.4, este nivel de energía de la señal es promediado sobre un tiempo de 8 periodos de símbolo que produce como resultado un valor que indica el nivel de energía como un entero de 8 bits dentro de un rango de 0 a 255.

El valor mínimo de ED (0) indica que la potencia recibida es inferior a 10 dB por encima de la sensibilidad especificada por el receptor. La sensibilidad del receptor está dada por el nivel de energía más bajo que el receptor puede detectar y demodular exitosamente con una tasa de error de menos del 1% en los paquetes recibidos.

El estándar define una sensibilidad mínima de recepción de -85 dBm para la banda de 2.4 GHz y una sensibilidad de -92 dBm para las bandas de 868 y 915 MHz. El estándar especifica que el rango de detección abarcado por los valores de ED (0 a 255) será de al menos 40 dB, es decir si el receptor posee una sensibilidad de -85 dBm su rango mínimo de detección será de -85 dBm a -45 dBm. Dentro de este rango, el mapeo de la potencia recibida en decibelios a los valores ED tendrá una precisión de ± 6 dB. [11]

➤ *Estructura del paquete de capa física*

La unidad de datos del protocolo del estándar 802.15.4 a nivel físico o PPDU (*Physical Protocol Data Unit*) encapsula todas las estructuras de datos de las capas superiores del estándar.

La PPDU consta de 3 componentes: un encabezado de sincronización o SHR (*Synchronization Header*), un encabezado físico o PHR (*PHY Header*) y la carga útil o PSDU (*PHY Service Data Unit*). Ver Figura 1.8.

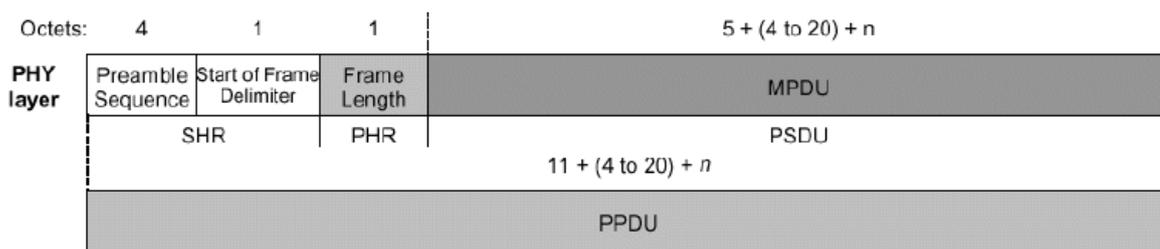


Figura 1.8. Estructura del paquete de capa física [16].

El campo SHR permite que el dispositivo receptor se sincronice y bloquee el flujo de bits. El campo PHR contiene información de la longitud del paquete de datos y el campo PHY, de longitud variable con un máximo de 127 bytes, almacena la información proveniente de las capas superiores. El formato completo de la PPDU de la capa física posee una longitud máxima de 133 bytes.

➤ *Convivencia de las redes que trabajan con el estándar IEEE 802.15.4 con otros estándares*

Convivencia con Bluetooth (Estándar IEEE 802.15.1)

El estándar IEEE 802.15.1 usa una técnica de modulación FHSS⁷ (*Frequency Hopping Spread Spectrum*) y además utiliza un mecanismo de salto de frecuencia adaptivo, el cual a través de un algoritmo cambia de forma dinámica la secuencia de saltos de frecuencia, logrando así limitar la cantidad de canales que se pueden usar, liberando ciertas frecuencias para que puedan ser usadas por otros estándares como IEEE 802.15.4 [36].

⁷ FHSS, por sus siglas en inglés "*Frequency Hopping Spread Spectrum*" es un tipo de modulación que consiste en el cambio repetido de la frecuencia portadora durante la transmisión de radio para reducir la interferencia y evitar la intercepción.

Convivencia con Wifi (Estándar IEEE 802.11)

El problema de la interferencia entre los estándares IEEE 802.11 e IEEE 802.15.4 se debe al solapamiento de ciertos canales. Sin embargo, se debe tomar en cuenta que existen cuatro canales (15, 20, 25, 26) en los que se disminuye la interferencia entre los dos estándares, dichos canales forman parte de la banda de frecuencia de los 2.4 GHz [36].

1.3.2.3.2 Subcapa MAC.

La subcapa de acceso al medio (MAC, *Medium Access Control*) provee una interfaz entre la capa física y la capa superior. El estándar IEEE 802.15.4 no provee una especificación para la capa superior, la cual por lo general es operada por protocolos como ZigBee, 6 LoWPAN, etc.

La capa MAC provee una interfaz con la capa superior mediante los siguientes dos servicios [8][13]:

- **Servicio de Administración MAC (MAC Management Service):** A través de un punto de acceso (SAP⁸) llamado MLME (MAC Layer Management Entity), este servicio es responsable de tareas de monitoreo y control (Administración).
- **Servicio de datos MAC (MAC Data Service):** usa un punto de acceso (SAP) llamado MCPS (MAC Common Port Layer), este servicio está encargado de la gestión de datos.

La Figura 1.9 muestra la interfaz entre la capa MAC y la capa superior.

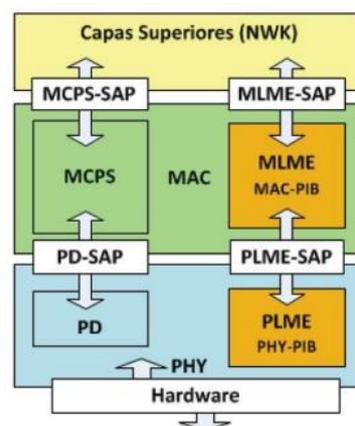


Figura 1.9. Interfaz entre capa MAC y la capa superior [19].

⁸ SAP, por sus siglas en inglés "Service Access Point" identifica un servicio concreto de un usuario que envía y recibe una clase de datos concreta.

La capa de acceso al medio también es responsable del control de flujo a través de la entrega de acuses de recibo, la validación de tramas, así como de mantener la sincronización de la red, controlar la asociación y planificar los mecanismos de acceso al canal. [14]

➤ *Mecanismos de acceso al medio*

El algoritmo CSMA-CA se utilizará antes de la transmisión de datos o tramas de comandos MAC transmitidas dentro del CAP. El estándar IEEE 802.15.4 utiliza dos tipos de mecanismos de acceso al canal, según la configuración de la red [17]:

- ✓ **CSMA-CA no ranurado:** las redes sin *beacon* utilizan este mecanismo de acceso al canal. Cada vez que un dispositivo desee transmitir tramas de datos o comandos MAC, deberá esperar un período aleatorio (*backoff*⁹). Si se encuentra que el canal está inactivo, luego del *backoff*, el dispositivo transmitirá sus datos. Si se encuentra que el canal está ocupado, luego del *backoff*, el dispositivo deberá esperar otro período aleatorio antes de intentar acceder al canal nuevamente [17].
- ✓ **CSMA-CA ranurado:** las redes habilitadas para *beacon* utilizan este mecanismo de acceso al canal, donde las ranuras de *backoff* se alinean con el inicio de la transmisión *beacon*. Cada vez que un dispositivo desea transmitir tramas de datos durante el CAP, debe ubicar el límite del siguiente intervalo de *backoff* y luego esperar un número aleatorio de intervalos de *backoff*. Si el canal está ocupado, después de este *backoff*, el dispositivo deberá esperar otro número aleatorio de intervalos de *backoff* antes de intentar acceder al canal nuevamente. Si el canal está inactivo, el dispositivo puede comenzar a transmitir en el siguiente límite de ranura de retroceso disponible [17].

➤ *Modalidades de Operación*

Modo beacon habilitado

En este modo se usa una supertrama para lograr la comunicación entre nodos. El formato de dicha supertrama es definida por el coordinador PAN y se envía dentro de una trama *beacon* periódicamente [18].

⁹ *Backoff* es tiempo aleatorio de espera al que deben atenerse los nodos tras haberse producido una transmisión fallida.

- Estructura de la supertrama

A un dispositivo se le puede asignar uno o más intervalos GTS para transferir tráfico de red, durante ese tiempo ningún otro dispositivo puede usar el canal. Cuando se usa un GTS, la red no usa el esquema de acceso al canal CSMA-CA, de ahí la frase "intervalo de tiempo garantizado". El período libre de contención siempre sigue al período de acceso a la contención, donde los dispositivos que no tienen una reserva previa para un intervalo de tiempo pueden usar el algoritmo CSMA-CA para acceder al canal y pasar su tráfico. Una vez que se completa el período sin contención, el dispositivo que transmite la trama *beacon* puede volverse inactivo, ahorrando energía hasta el comienzo del siguiente intervalo de *beacon* [16].

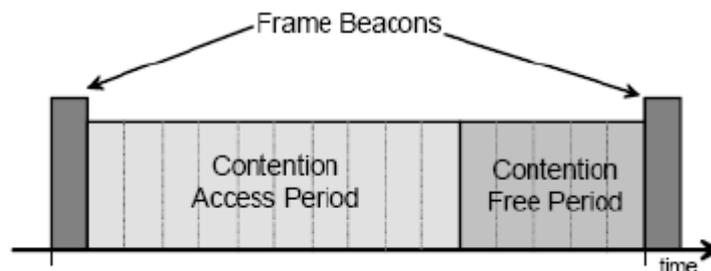


Figura 1.10. Estructura de la supertrama [16].

Modo beacon no habilitado

En este modo los nodos pueden enviar su información mediante el mecanismo de acceso al medio CSMA-CA no ranurado, en este modo de operación no se usan supertamas [18].

➤ *Modalidades de transferencia de información*

Los mecanismos para la transferencia de datos dependen de si la red admite la transmisión de *beacons*. Se utiliza una red habilitada para *beacons* para admitir dispositivos de baja latencia, como periféricos de PC. Si la red no necesita admitir dichos dispositivos, puede optar por no utilizar la beacon para las transferencias normales [17].

- **Transmisión directa de datos:**

Esta transferencia de datos es el mecanismo para transferir datos desde un dispositivo a un coordinador. En una red habilitada para *beacons*,

cuando un dispositivo desea transferir datos a un coordinador, primero escucha la *beacon* de la red, como se muestra en la Figura 1.11. Cuando se encuentra la *beacon*, el dispositivo se sincroniza con la estructura de la supertrama. En el punto apropiado, el dispositivo transmite su trama de datos, usando CSMA-CA ranurado, al coordinador. El coordinador reconoce la recepción exitosa de los datos mediante la transmisión de una trama ACK [17].

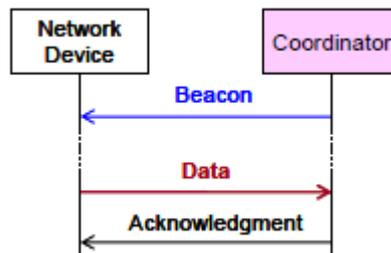


Figura 1.11. Transferencia de datos de dispositivo a coordinador con modo *beacon* habilitado [17].

Por otro lado, en una red sin *beacon*, cuando un dispositivo desea transferir datos, simplemente transmite su trama de datos, utilizando CSMA-CA no ranurado, al coordinador. El coordinador reconoce la recepción exitosa de los datos mediante la transmisión de una trama ACK, como se muestra en la Figura 1.12 [17].

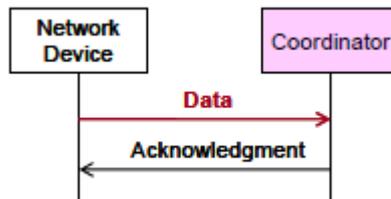


Figura 1.12. Transferencia de datos de dispositivo a coordinador con modo beacon no habilitado [17].

- **Transmisión indirecta de datos:**

Esta transferencia de datos es el mecanismo para transferir datos de un coordinador a un dispositivo. En una red habilitada para *beacons*, cuando el coordinador desea transmitir datos a un dispositivo, indica en la *beacon* de la red que el mensaje de datos está pendiente. El dispositivo escucha periódicamente la *beacon* de la red y, si hay un mensaje pendiente,

transmite un comando MAC solicitando los datos, utilizando CSMA-CA ranurado. El coordinador reconoce la recepción exitosa de la solicitud de datos mediante la transmisión de una trama ACK. A continuación, la trama de datos pendiente se envía mediante CSMA-CA ranurado. El dispositivo reconoce la recepción exitosa de los datos mediante la transmisión de una trama ACK. Al recibir el acuse de recibo, el mensaje se elimina de la lista de mensajes pendientes en el *beacon*. Este proceso se muestra en la Figura 1.13.

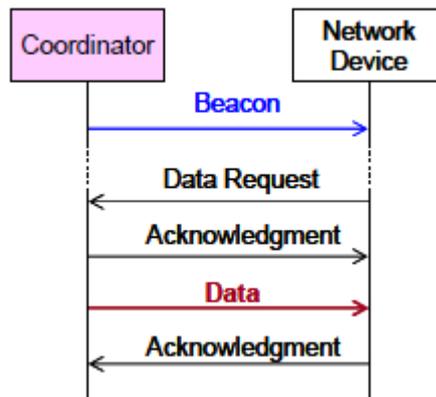


Figura 1.13. Transferencia de datos de coordinador a dispositivo con modo beacon habilitado [17].

En una red sin *beacon*, un dispositivo puede establecer contacto mediante la transmisión de un comando MAC solicitando los datos, utilizando CSMA-CA no ranurado, a su coordinador a una tasa de sondeo definida por la aplicación. El coordinador reconoce la recepción exitosa de la solicitud de datos mediante la transmisión de una trama ACK. Si hay datos pendientes, el coordinador transmite la trama de datos, utilizando CSMA-CA sin ranura, al dispositivo. Si no hay datos pendientes, el coordinador transmite una trama de datos con una carga útil de longitud cero para indicar que no hay datos pendientes. El dispositivo reconoce la recepción exitosa de los datos mediante la transmisión de una trama de reconocimiento, como se muestra en la Figura 1.14.

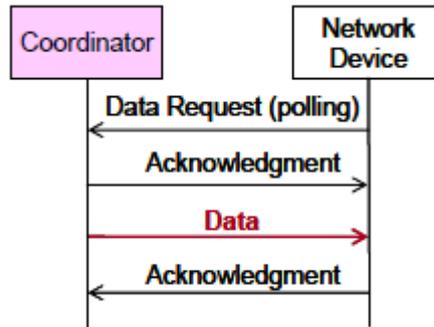


Figura 1.14. Transferencia de datos de coordinador a dispositivo con modo *beacon* no habilitado [17].

➤ *Estructura de las tramas MAC*

La estructura de la trama MAC posee los elementos necesarios para superar los desafíos de un medio inalámbrico. El estándar 802.15.4 define cuatro estructuras de trama:

✓ **Trama de datos**

Está compuesta de la siguiente manera [8].

- **Cabecera MAC** (MHR, *MAC Header*), dividida por los siguientes campos:
 - Campo de control de trama (FCF, *Frame Control field*). Indica el tipo de trama que se va a transmitir, posee una longitud de 2 bytes.
 - Campo de control de secuencia. Identifica el número de secuencia de la trama, posee una longitud de 1 byte.
 - Campo de direccionamiento. Posee información del direccionamiento de la trama, es un campo de longitud variable de 0 a 20 bytes.
- **Unidad de servicios de datos MAC (MSDU, *MAC Service Data Unit*)**, campo de longitud variable que contiene la carga útil de la trama, lleva información específica a transmitirse. En conjunto con la trama MAC completa su tamaño no debe superar los 127 bytes.
- **Fin de Trama o MFR (*MAC Footer*)**, campo que contiene la secuencia de chequeo de la trama (FCS, *Frame Check Sequence*) que permite la

detección de errores y ayuda a verificar la integridad de la trama. Posee una longitud de 2 bytes.

En la Figura 1.15 se puede observar el formato general de una trama MAC IEEE 802.15.4

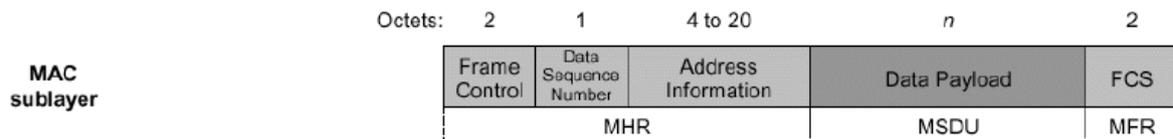


Figura 1.15. Formato de la trama de datos [16].

✓ **Trama ACK (*Acknowledgment*)**

Los campos diferentes a la trama de datos son los siguientes [18]:

- *Sequence Number*, hace referencia al número de secuencia de la trama que se ha recibido.

En la Figura 1.16 se puede observar el formato general de una trama ACK.

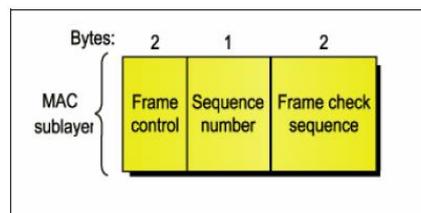


Figura 1.16. Formato de la trama ACK [18].

✓ **Trama de comandos**

Los campos diferentes a la trama de datos son los siguientes [18]:

- *Command Type*, hace referencia al tipo de comando generado.
- *Command Payload*, contiene información del tipo de comando generado.

En la Figura 1.17 se puede observar el formato general de una trama de comandos MAC.

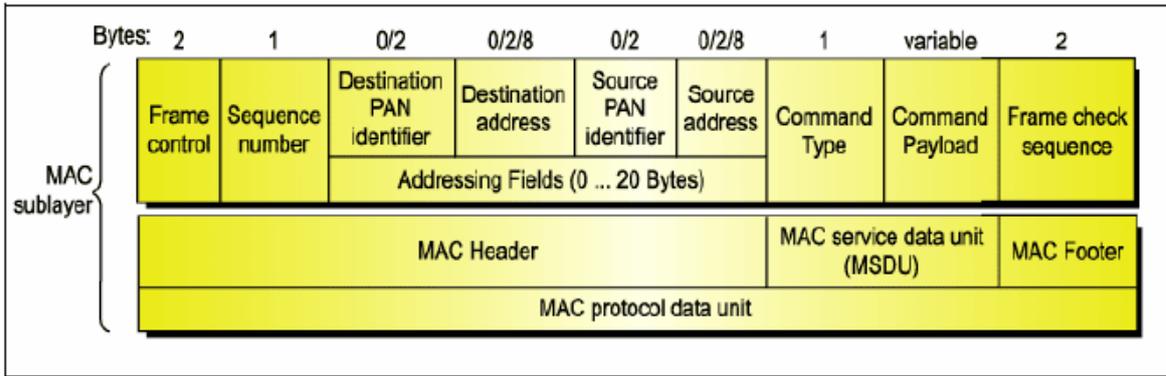


Figura 1.17. Formato de la trama de comandos [18].

✓ **Trama beacon**

Los campos diferentes a la trama de datos son los siguientes [18]:

- *Superframe Specification*, hace referencia a los parámetros de la supertrama.
- GTS Fields, posee información referente a las GTS asignados por el coordinador.
- *Pending Address*, contiene datos sobre los dispositivos que tienen información por enviar al coordinador.
- *Beacon Payload*, posee información que viene de las capas superiores.

En la Figura 1.18 se puede observar el formato general de una trama *beacon*.

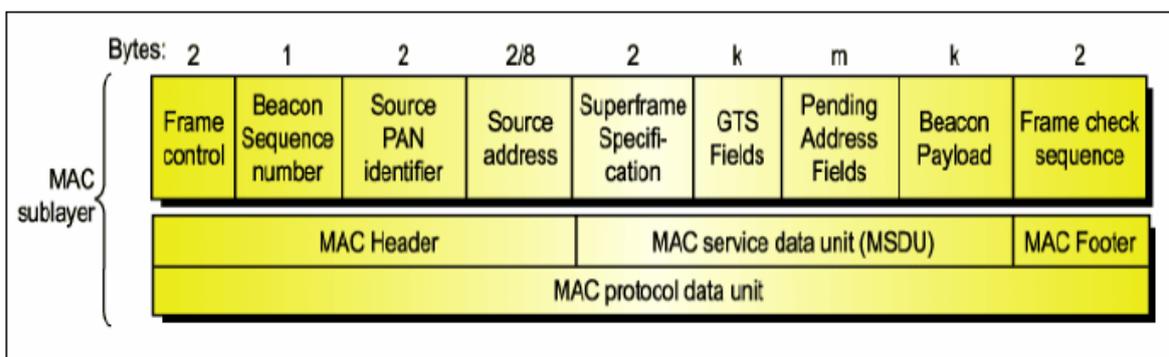


Figura 1.18. Formato de la trama *beacon* [18].

➤ *Direccionamiento en el estándar IEEE 802.15.4*

El estándar sugiere que todas las direcciones se basen en valores únicos de 64 bits (Dirección extendida). Sin embargo, para conservar el ancho de banda y reducir el consumo de energía en la transmisión, el estándar también soporta el uso de direcciones cortas de 16 bits. Además, para reducir aún más el uso de bits en el campo de direccionamiento del encabezado MAC(MHR), el identificador PAN (PAN ID) origen o destino también se puede obviar para permitir una transmisión más eficiente.

Dentro del encabezado de la trama MAC se indica el tipo de direccionamiento que se está usando, tanto para el origen como para el destino de la trama enviada. En la figura se muestra cómo se encuentran las direcciones dentro de la cabecera de una trama IEEE 802.15.4 [15] [20].



Figura 1.19. Información de direccionamiento dentro del encabezado de trama [15].

- **Dirección extendida (*extended address*):** es una dirección de 64 bits, asignada por la IEEE, que identifica de manera única al dispositivo; no hay otro dispositivo en el mundo que pueda tener la misma dirección. A veces también llamada dirección IEEE [21].
- **Dirección corta (*short address*):** esta dirección de 16 bits identifica a un nodo en la red y es tanto local como global para esa red (por lo que dos nodos en diferentes redes pueden tener la misma dirección corta). El coordinador es el encargado de asignar direcciones cortas cuando los nodos se unan a la red. Si el dispositivo no tiene una dirección corta, debe direccionarse utilizando su dirección IEEE [21].
- **Identificador de red (PAN ID):** permite identificar tanto la red de destino como la red de origen en el proceso de envío y recepción de tramas. Existe una bandera

(*flag*) ubicada dentro del encabezado, que permite emplear solo el PAN ID de destino, para de esta manera poder enviar tráfico dentro de una misma red PAN (intra-pan) [15].

- **Broadcast:** permite enviar información útil a todos los dispositivos pertenecientes a una misma red PAN, así como a diferentes redes PAN. Para enviar un *broadcast* dentro de una red PAN se usa una dirección de destino corta con valor 0xFFFF, en caso de requerir el envío de *broadcast* a otras redes PAN se debe colocar el valor de 0xFFFF en el campo PAN ID [15].

1.3.3 ZigBee

ZigBee es una tecnología inalámbrica de capa superior creada en base al estándar IEEE 802.15.4, su objetivo es proveer una interconexión simple a nodos inalámbricos de bajo consumo de potencia y baja capacidad de procesamiento.

La especificación ZigBee es mantenida y actualizada por la alianza ZigBee (*ZigBee Alliance*), la cual es la encargada de definir la red, la seguridad y la capa de aplicación de la misma, mientras que el estándar IEEE 802.15.4 define su capa física y capa de acceso al medio [22]. Ver Figura 1.20.

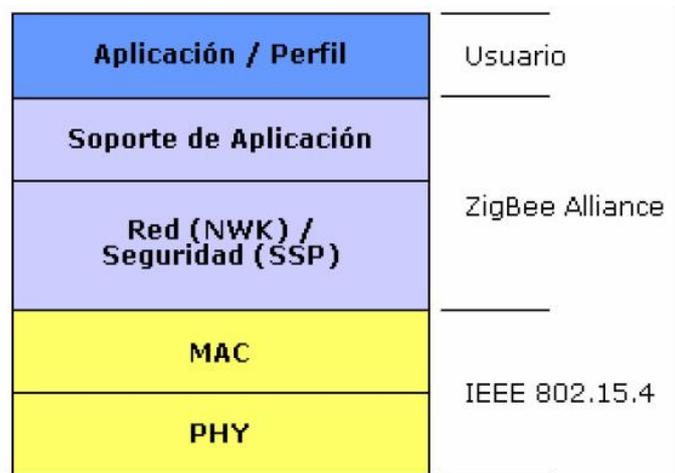


Figura 1.20. Protocolo ZigBee[41].

1.3.3.1 Tipos de dispositivos ZigBee

ZigBee define los siguientes dispositivos [23]:

- **Coordinador:** dispositivo FFD encargado de la gestión general de la red. Cada red ZigBee solo posee un dispositivo coordinador. El coordinador es responsable de las siguientes funciones: seleccionar el canal inalámbrico, iniciar la red, asignar

direcciones a los nodos y enrutadores, permitir que otros dispositivos se unan o abandonen la red, transferir paquetes de aplicaciones y ser la puerta de enlace o Gateway para la comunicación con redes externas.

- **Dispositivo Final:** dispositivo RFD que opera con una suficiente funcionalidad de la capa MAC IEEE 802.15.4 para comunicarse con los dispositivos enrutadores o coordinadores. Son dispositivos de baja potencia y bajo costo.
- **Enrutador:** dispositivo FFD usado para expandir la cobertura de la red ZigBee. Actúa como nodo intermediario cuya funcionalidad principal es encontrar las mejores rutas para retransmitir la información a los nodos finales.

1.3.3.2 Topologías ZigBee

Las topologías ZigBee se basan en las topologías IEEE 802.15.4 explicadas en la sección 1.3.2.2. Si bien el estándar IEEE 802.15.4 soporta las topologías: estrella, malla (llamada también punto a punto), árbol y *cluster tree*; ZigBee solo soporta las topologías estrella, malla y árbol. [22] [23]

ZigBee está dirigido a trabajar en topología de estrella o árbol. Las topologías de malla pueden resultar interesantes, pero debido a que ZigBee solo permite que los dispositivos finales activen el modo suspensión, no se puede tener una topología de malla real sin tener un exceso de consumo de energía. Para trabajar en una topología de malla pura, se recomienda elegir otro protocolo como *DigiMesh*¹⁰ [24].

1.3.3.3 Direccionamiento en topologías lineales ZigBee

La asignación de direcciones o identificadores a los nodos puede ser realizada automáticamente aplicando protocolos de enrutamiento definidos por ZigBee. ZigBee ofrece los siguientes mecanismos de direccionamiento los cuales son ineficientes o simplemente no pueden ser implementados en topologías lineales [25] [26].

- **Mecanismo de asignación de direcciones estocásticas (SAAM, *Stochastic Address Assignment Mechanism*):** este mecanismo de direccionamiento verifica la duplicidad de las direcciones después de asignar identificadores aleatorios a los nodos. Es usado en redes con un alto número de nodos inalámbricos presentes; sin embargo, existe una alta probabilidad de generar

¹⁰ *DigiMesh*: es un protocolo de comunicación inalámbrico propietario de *Digi International*, el cual es prácticamente una versión mejorada de Zigbee simplificando la arquitectura y la interconexión entre los nodos de la red.

conflicto en la asignación de direcciones, por lo cual es necesario el uso del protocolo AODV¹¹ para resolver dichos conflictos. Debido a los conflictos mencionados este mecanismo no puede ser implementado en redes multisalto lineales a gran escala.

- **Mecanismo de Asignación de Direcciones Distribuidas (DAAM, *Distributed Address Assignment Mechanism*):** mecanismo de direccionamiento tipo padre-hijo que forma una topología tipo cluster tree. DAAM organiza todos los nodos en la red y enruta paquetes utilizando información de direccionamiento sin la necesidad de usar tablas de enrutamiento adicionales. En redes lineales de sensores este mecanismo no es aplicable cuando se usa 802.15.4 dado que los nodos tienen poca cobertura.
- **Esquema de direccionamiento *Long – Thin*:** ZigBee propone este mecanismo de asignación el cual está destinado a usarse en redes inalámbricas que poseen múltiples trayectorias lineales entre dos nodos, conocidas como redes LT-WSN (*Long Thin Wireless Sensor Network*). Las redes LT-WSN están divididas en clústeres, en cada clúster dos nuevos roles son definidos: un nodo cabeza de grupo y un nodo puente. Estos nuevos roles tienen que ser manualmente seleccionados entre todos los nodos de la red para iniciar la asignación de direcciones. Si el nodo puente falla, el enrutamiento deja de funcionar. En la Figura 1.21 se puede observar este tipo de red, los nodos entre el nodo cabecera de grupo y nodo puente son nodos básicos. Dado que este mecanismo requiere la configuración manual de los roles no es eficiente su utilización en redes lineales multisalto.

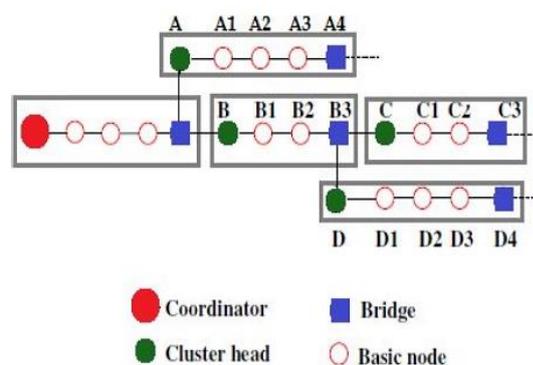


Figura 1.21. Red inalámbrica de sensores *Long-Thin* [25].

¹¹ AODV, por sus siglas en inglés “*Ad hoc On-Demand Distance Vector*” es un protocolo de enrutamiento para redes ad hoc móviles.

1.3.4 6LoWPAN

El estándar 6LoWPAN fue desarrollado por la IETF (*Internet Engineering Task Force*) con el objetivo de transmitir paquetes IPV6¹² sobre el estándar IEEE 802.15.4, para de esta manera aprovechar las ventajas del direccionamiento de IPV6 y las características de bajo consumo y procesamiento de IEEE 802.15.4.

Entre los beneficios de 6LoWPAN se encuentran: facilidad de uso al ser un estándar abierto, integración transparente con Internet, escalabilidad global, flujo end-to-end, interoperabilidad con otras tecnologías de internet, etc.

La Figura 1.22 muestra la arquitectura del protocolo 6LoWPAN. En comparación con la arquitectura del protocolo TCP/IP¹³ de Internet, 6LoWPAN agrega una nueva capa adicional llamada capa de adaptación.

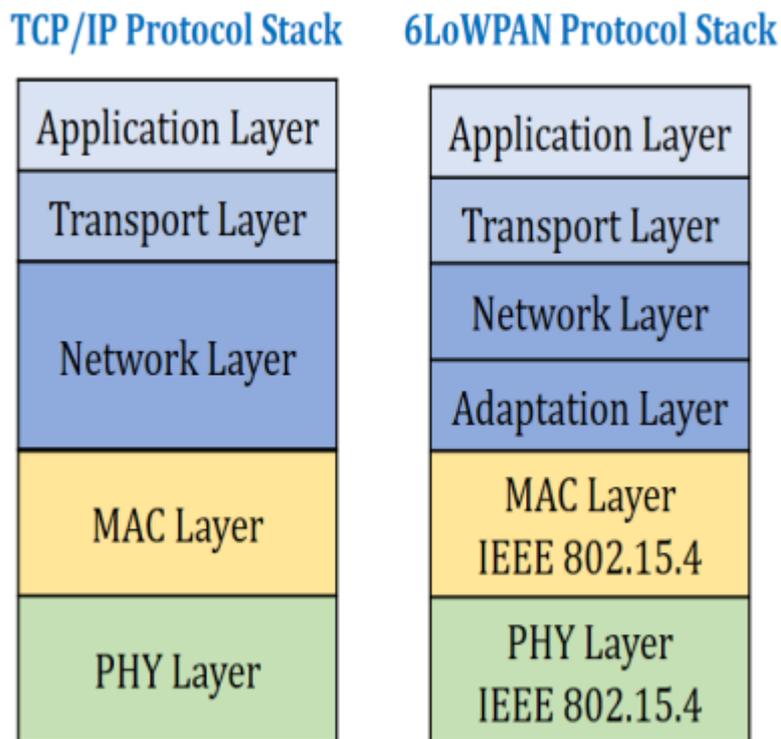


Figura 1.22. Arquitectura del protocolo TCP/IP y 6LoWPAN [27].

La capa de adaptación provee mecanismos de fragmentación y reensamblaje que permiten a las capas inferiores descomponer los paquetes IPV6 con un MTU (Unidad

¹² IPv6 es una actualización al protocolo IPv4, diseñado para resolver el problema de agotamiento de direcciones.

¹³ TCP/IP por sus siglas en inglés "*Transmission Control Protocol/Internet Protocol*" es un conjunto de reglas estandarizadas que permiten a los equipos comunicarse en una red como Internet.

Máxima de Transmisión) de 1280 bytes, en fragmentos más pequeños de 127 bytes permitidos por el estándar IEEE 802.15.4, para que mediante este mecanismo los paquetes puedan ser transportados por las capas inferiores de manera transparente a IPv6. Además, la capa de adaptación proporciona compresión de cabeceras cuando la carga útil de la MAC IEEE 802.15.4 no pueda ajustarse al paquete IPv6.

6LoWPAN soporta las topologías tipo estrella y malla del estándar IEEE 802.15.4, opera comúnmente en redes multisalto formando una red tipo malla de bajo consumo [28].

1.3.4.1 Direccionamiento en 6LoWPAN

El estándar usa una dirección de origen auto configurable para asignar direcciones a cada nodo en la red, mediante la generación de un enlace local que usa una dirección extendida o corta del estándar IEEE 802.15.4. El alcance de este tipo de direcciones permite cubrir toda una red 6LoWPAN, incluso sobre una red de múltiples saltos; sin embargo, se necesita una dirección enrutable cuando los dispositivos están ubicados a múltiples saltos de distancia.

Para obtener una dirección de enlace local un dispositivo 6LoWPAN hace uso del protocolo *Neighbor Discovery* (ND), el cual permite al dispositivo encontrar nodos vecinos, configurar rutas, propagar configuración de la red y mantener información acerca de la disponibilidad de sus nodos vecinos [29].

1.3.5 MÓDULO TRANSECTOR ZIGBIT DE ATMEL

La placa *Zigbit*® de desarrollo es una extensión del kit *XPLAINED PRO* de *Atmel*, esta placa lleva el nombre de serie ATZB-256RFR2-XPRO [30]. El principal componente de esta placa de desarrollo es el módulo ATmega256RFR2 *ZigBit*, que en un solo chip contiene un microcontrolador AVR¹⁴ de 8 bits y un transceptor de alta velocidad de datos para la banda ISM de 2,4 GHz compatible con el estándar ZigBee o IEEE 802.15.4. El transceptor de radiofrecuencia proporciona altas velocidades de datos desde 250 Kb/s hasta 2 Mb/s, manejo de tramas, excelente sensibilidad del receptor y alta potencia de salida de transmisión que permite una comunicación inalámbrica muy robusta.

El módulo está diseñado para soportar aplicaciones inalámbricas de detección, monitoreo, control y adquisición de datos [30].

¹⁴ AVR se refiere a la familia de microcontroladores RISC del fabricante estadounidense *Atmel*, compañía adquirida por *Microchip Technology*.

Las principales características del módulo *ZigBit®* Atmega256RFR2 se presentan a continuación [31]:

- Sensibilidad de recepción de -97 dBm
- Presupuesto de enlace de hasta 100,6 dB
- Potencia de salida de hasta +3,6 dBm
- Memoria de programa flash no volátil de 256 KB, memoria EEPROM de 8 KB y memoria SRAM¹⁵ de 32 KB
- Amplia gama de interfaces (tanto analógicas como digitales)
 - ISP (*In-System Programming*, Programación en el sistema), *JTAG*
 - 2 entradas de comparadores analógicos
 - Conexión Serial tipo UART y USART
 - Temporizador, PWM (Modulación por ancho de pulsos)
 - 4 líneas ADC (Conversor Analógico-Digital)
 - Entrada de reloj externo, salida de reloj interno
 - Hasta 31 líneas configurables como GPIO (*General Purpose Input Output*, Pines de entrada/salida de propósito general)
- Velocidades de transmisión desde 250 kb/s hasta 2 Mb/s
- Antena integrada en el chip

1.3.5.1 Características de la placa de desarrollo Zigbit

Los componentes de la placa ATZB-256RFR2-XPRO se pueden observar en la Figura 1.23.

¹⁵ SRAM, por sus siglas en inglés "*Static Random Access Memory*" es un tipo de memoria RAM usado en varias aplicaciones electrónicas, solo mantiene su contenido mientras se le está aplicando una fuente de alimentación.

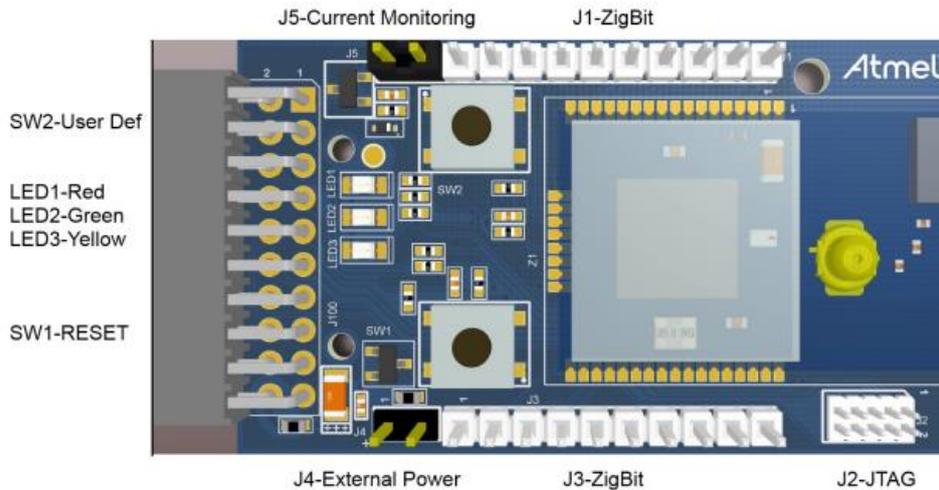


Figura 1.23. Vista general de la placa de desarrollo ZigBit ATZB-256RFR2-XPRO [32].

El funcionamiento tanto de los botones y puertos se explicarán a continuación.

1.3.5.1.1 Botones y LEDs

Dependiendo de la aplicación y mediante su configuración oportuna, los botones y LEDs pueden ser usados a conveniencia del usuario [32].

- **LED:** El módulo nos permite usar 3 LED. LED1-Red, LED2-Green, LED3-Yellow los cuales están conectados a los puertos PD6, PG2 y PE2 respectivamente.
- **Botones:** El módulo nos permite usar dos botones. Un botón SW1 que permite reiniciar el microcontrolador y un botón SW2 de propósito general que puede ser programado por el usuario. El botón genera una señal de bajo voltaje cuando es presionado.

1.3.5.1.2 Programación de interfaces

Para programar la placa se hace uso de su interfaz JTAG ubicada en el puerto J2, esto sobrescribirá el programa instalado en el *bootloader* o gestor de arranque. Su conexión se puede apreciar en la Figura 1.24.



Figura 1.24. Conexión de la interfaz JTAG para programar la placa [32].

1.3.5.1.3 Fuente de alimentación

El puerto J4 habilita la conexión de una fuente de alimentación externa de 3.3V, típicamente usado con dos pilas AAA, este puerto tiene una protección de voltaje inverso. La conexión de las pilas se puede observar en la Figura 1.25.



Figura 1.25. Conexión de la fuente de alimentación a la placa de desarrollo [32].

1.3.5.1.4 Pines ZigBit

Los puertos J1 y J3 que pueden ser observados en la Figura 1.23 nos brindan acceso a todos los pines, estos pines de entrada y salida son de propósito general y son usados según el tipo de aplicación que se requiera. Además, permiten la conexión a otra placa principal.

1.3.6 HERRAMIENTA DE DESARROLLO DE ATMEL.

Atmel Studio 7 es una plataforma de desarrollo integrada (IDP) que permite el desarrollo y depuración de código para aplicaciones que se ejecutan en sistemas embebidos con microcontroladores AVR®, brinda un entorno ligero y sencillo para escribir, crear y depurar aplicaciones escritas en C/C++ o código ensamblador. También interactúa de manera perfecta con depuradores y kits de desarrollo de *Atmel*. Además, *Atmel Studio* incluye *Gallery*, una tienda de aplicaciones en línea que permite ampliar el entorno de desarrollo con herramientas tanto nativas como de terceros.

Atmel también provee un marco de trabajo avanzado ASF (*Atmel Software Framework*) el cual contiene una amplia librería de códigos fuente, controladores, paquetes de comunicación y una gran variedad de proyectos de ejemplo los cuales permiten agilizar el desarrollo de aplicaciones. [40]

Así mismo, *Atmel* proporciona la extensión llamada *Wireless Composer* la cual permite crear proyectos inalámbricos compatibles con transceptores que operan con el estándar IEEE 802.15.4, mediante el uso de la librería de software AVR®2025 MAC desarrollada por *Atmel*.

1.3.6.1 Arquitectura AVR2025 MAC de Atmel

La librería AVR®2025 MAC de *Atmel* soporta una implementación completa del estándar IEEE 802.15.4 y está escrita para admitir la familia de microcontroladores de *Atmel*. La librería consta de tres módulos o capas principales las cuales permiten su uso según el requerimiento funcional de cada aplicación [33].

Las capas principales son: PAL, TAL, MCL. Ver Figura 1.26.

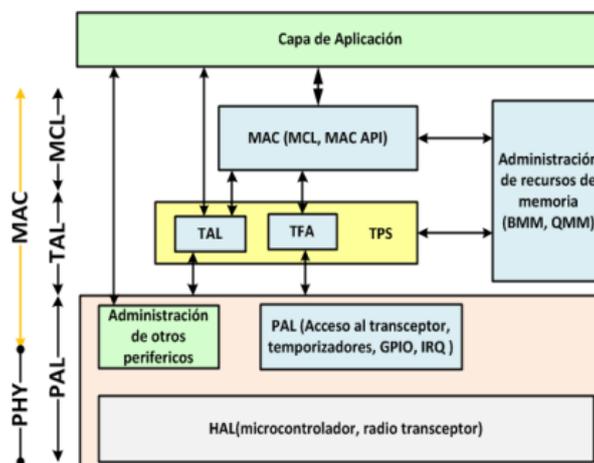


Figura 1.26. Arquitectura por bloques que utiliza Atmel [15].

1.3.6.1.1 Capa PAL

El objetivo de la capa de abstracción de plataforma PAL (*Platform Abstraction Layer*) es separar la funcionalidad de un microcontrolador específico y proporcionar una interfaz común para acceder a los diferentes modelos. La separación adecuada de la capa PAL del resto de la pila de *software* es importante ya que permite mantener las capas superiores independientes de los detalles del *hardware*, además permite una fácil migración a otros tipos de chips [33].

La capa PAL contiene interfaces a los siguientes componentes [33].:

- Acceso a la funcionalidad del transceptor.
- Manejo GPIO (*General Purpose Input/Output*, Pines de propósito general de entrada y salida),
- Manejo de interrupciones de bajo nivel.
- Temporizadores (Temporizadores de alta prioridad y Temporizadores de *software*)
- Manejo de entradas/salidas de flujo a través de USB.
- Manejo de botones y LEDs.
- Acceso a memorias FLASH o EEPROM.
- IRQ (*Interrupt Request*, Peticiones de Interrupción), mecanismo para inicializar y controlar las interrupciones de *hardware* mediante subrutinas.

1.3.6.1.2 Capa TAL

La capa de abstracción del transceptor TAL (*Transceiver Abstraction Layer*), como sugiere su nombre, es responsable de proporcionar una interfaz para la funcionalidad del transceptor sin preocuparse por el transceptor subyacente específico [33].

La capa TAL contiene los siguientes componentes [33]:

- Unidad de transmisión de tramas.
- Unidad de recepción de tramas.
- Almacenamiento TAL PIB (Base de datos con información de la red PAN en la capa TAL)
- Módulo que implementa el mecanismo CSMA-CA

- Detección de Energía.
- Manejo de interrupciones.
- Inicialización y reinicio
- Administración de la energía

La capa TAL satisface los requerimientos y características del estándar IEEE 802.15.4, en contraste *Atmel* proporciona otro bloque de software llamado TFA (*Transceiver Feature Access*) que proporciona funcionalidades extra al *hardware* subyacente.

Funcionalidades implementadas en el bloque TFA:

- Manejo adicional de los atributos de la PIB, ejemplo: manipulación de la sensibilidad del transceptor en la recepción de tramas.
- Manejo CCA (*Clear Chanel Assessment*, Evaluación del canal libre).
- Lectura del voltaje del transceptor.
- Medición de temperatura del dispositivo.

1.3.6.1.3 Capa MCL

La capa núcleo de MAC (*MAC Core Layer*) proporciona una interfaz para una capa de red compatible con el estándar IEEE 802.15.4. Es responsable de proveer servicios como [33]:

- Despachador de MAC.
- Servicio de datos MAC.
- Servicio de administración MAC (asociación, escaneo, sondeo, etc).
- Manejo de tramas MAC *beacon* .
- Módulo MAC PIB.

2. METODOLOGÍA

Para el desarrollo del presente proyecto se hizo uso de una investigación aplicada y experimental, implementando un algoritmo para la asignación de direcciones en una topología lineal usando el estándar IEEE 802.15.4, el cual fue probado en un prototipo conformado por los módulos ATZB – 256RFR2 de *Atmel* que contienen las herramientas necesarias para manipular y trabajar sobre el estándar IEEE 802.15.4.

2.1. DISEÑO DEL ALGORITMO

2.1.1. REQUERIMIENTOS DEL ALGORITMO

El algoritmo propuesto hace uso del estándar IEEE 802.15.4 operando en la banda de frecuencia de 2.4 GHz, utilizando CSMA/CA no ranurado para la transmisión y recepción de datos; y contemplando un esquema de direccionamiento corto descrito en el estándar, cada nodo en la red ejecuta el algoritmo para obtener automáticamente un identificador de red.

El algoritmo es implementado a un nivel de capa de enlace para evaluar sus ventajas al no usar capas superiores. Además, en la transmisión y recepción de datos no hace uso de acuses de recibo, reduciendo de mejor manera la sobrecarga en el control de la red.

El algoritmo considera una topología lineal multisalto con al menos seis nodos presentes en la red. Cada nodo puede comunicarse con al menos dos nodos a su izquierda o con dos nodos a su derecha, considerando un solo sentido en el proceso de asignación de direcciones.

Para garantizar una correcta asignación de direcciones, el algoritmo en su proceso considera enlaces fallidos y la presencia de nodos averiados en la red. Brindando de esta manera un grado de confiabilidad al proceso, en diferentes escenarios de fallo de la red.

2.1.2. DESCRIPCIÓN GENERAL DEL ALGORITMO

Cada nodo posee las siguientes configuraciones iniciales:

- Un identificador de la red PAN (0xCAFE) que permite distinguir a cada nodo participante en el proceso de asignación de direcciones y establecer comunicación solo con los nodos pertenecientes a la misma red.

- Una dirección corta aleatoria que permite a cada nodo poseer una dirección temporal inicial para brindarle la capacidad de transmitir y recibir datos al inicio del proceso de asignación de direcciones.

En la Figura 2.1 se pueden apreciar las configuraciones iniciales.

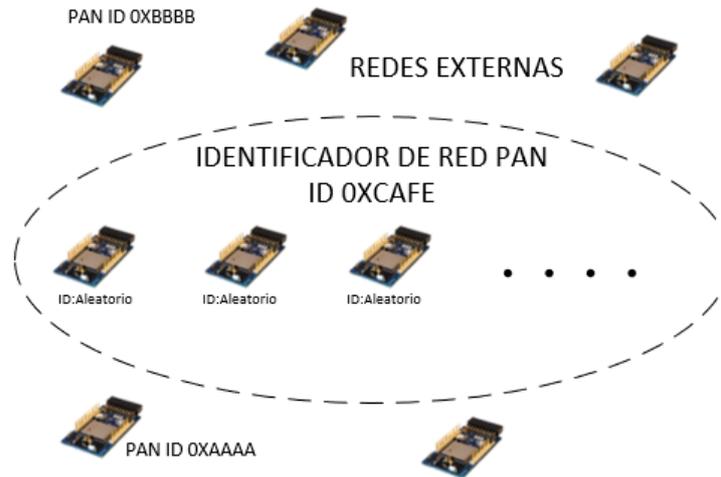


Figura 2.1. Configuraciones de red iniciales.

Se definen dos tipos de nodos: nodos asignadores y nodos simples o no asignadores; actuando como nodos FFD (*Full Function Device*) y nodos RFD (*Reduced Function Device*) respectivamente. Los roles y procesos generales de cada nodo se describen brevemente a continuación:

- **Nodo Asignador**

- Emite una señal de *broadcast* para detectar los nodos que requieran un identificador en la red.
- Almacena la información de direccionamiento y el nivel de potencia de la señal recibida dada por los nodos cercanos en la red para su procesamiento.
- Procesa la información recibida por los nodos no asignadores para actuar en tres distintos escenarios: asignación normal, enlaces fallidos y nodos caídos.
- Determina al nodo más alejado en su zona de cobertura para ser configurado como el próximo nodo asignador en la red.
- Asigna direcciones globales a los nodos en la red.

- **Nodo Simple**

- Responde a la señal de *broadcast* emitida por el nodo asignador en caso de requerir un identificador
- Recibe información de su direccionamiento y próximo rol en la red.
- Emite trama de comprobación para verificar su correcta asignación.

Con lo mencionado anteriormente la propuesta define a un nodo asignador el cual será responsable de proveer los identificadores a los nodos vecinos (vn) que requieran direcciones en la red. Para empezar el proceso, el nodo asignador inicial será automáticamente configurado con un identificador previamente definido (Id); el algoritmo automáticamente determinará cuáles serán los próximos nodos asignadores durante el proceso de asignación.

Los nodos asignadores usan el nivel de potencia de la señal recibida para identificar a los nodos de manera secuencial según su grado de proximidad física. Dado que la posición de los nodos y la distancia entre ellos permanece constante, la asignación secuencial de direcciones también puede brindar información estimada de la ubicación del nodo en una determinada área de cobertura.

El nodo asignador fija una dirección $Id + 1$ al nodo más cercano y un identificador $Id + x$ al nodo más alejado de la red, siendo "x" el identificador más alto o el número de nodos que el nodo asignador espera encontrar dentro de su área de cobertura. Para seleccionar los nodos cercanos y los nodos más alejados de la zona de cobertura, el algoritmo toma en cuenta lo siguiente:

1. En caso de producirse una asignación normal o una asignación con enlaces fallidos, el nodo asignador ordena de mayor a menor la información del nivel de señal recibida, para determinar los nodos más cercanos y al nodo más alejado; y poder enviar de manera secuencial los identificadores. El nodo asignador envía un identificador $id + 1$ al nodo con mayor nivel de potencia y un identificador $id + x$ al nodo con menor nivel de potencia.
2. En caso de producirse una asignación con nodos fallidos, dado que un nodo fallido puede ser un nodo cercano o un nodo alejado, es imposible mantener una identificación secuencial en la red usando un método de ordenamiento simple, ya que se podría confundir a los nodos dentro del área de cobertura. Para asignar identificadores en caso de existir nodos fallidos y no perder la secuencialidad de la red se usan umbrales de potencia, un nodo puede considerarse el más cercano o

el más lejano de la zona de cobertura del nodo asignador, si su nivel de señal se encuentra por encima o por debajo de un nivel de referencia. Por lo tanto, es necesario encontrar valores de referencia adecuados mediante una medición previa del nivel de potencia de la señal recibida.

El nodo asignador establecerá un identificador el cual se encuentra en el rango de direccionamiento corto de 16 bits definido por el estándar, dando un total de 65,536 (2^{16}) identificadores disponibles en la red.

Una vez el nodo asignador envíe los identificadores de red a cada nodo presente en su zona de cobertura, habrá seleccionado al próximo nodo asignador el cual se encargará de repetir el proceso de asignación tomando los siguientes nodos de red, formando así una topología lineal multisalto. El proceso se repetirá hasta que la comunicación alcance al último nodo de frontera o se encuentren al menos dos nodos averiados en la red.

2.1.2.1. Funcionamiento normal del algoritmo

En un escenario ideal, donde no se produzcan fallas en la comunicación y cada nodo en la red funcione adecuadamente, el algoritmo ejecuta el proceso de asignación sin la necesidad de realizar tareas adicionales que garanticen la correcta asignación de direcciones.

Antes de iniciar el proceso de asignación de direcciones cada nodo posee un identificador de red PAN (0xCAFE), una dirección corta aleatoria y tiene configurada la cantidad de nodos vecinos (v_n) que espera encontrar dentro de su zona de cobertura.

En cada envío y recepción de datos se ocupa parte del *payload* de la MPDU (Unidad de datos del protocolo de acceso al medio) con información o etiquetas que permiten a los nodos definir que tareas debe realizar. Cabe recalcar que el proceso de asignación de direcciones se ejecuta una sola vez y al presionar tres veces el botón SW2 de cualquier nodo transceptor elegido como nodo asignador.

A continuación, se presenta el proceso que se lleva a cabo después de presionar el botón:

1. El nodo asignador inicial v_0 es autoasignado con un identificador inicial (id) previamente configurado. Este nodo da inicio al proceso de direccionamiento de los demás nodos en la red. El nodo envía una trama *broadcast* con la etiqueta "A" para detectar los nodos que requieren identificadores y procede a esperar las tramas de respuesta activando un temporizador (*timer1*).

2. Los nodos vn que reciban el *broadcast* de direccionamiento y requieran un identificador responden al nodo asignador con una trama de respuesta con la etiqueta "R". La trama de respuesta tiene como dirección origen una dirección aleatoria y como dirección destino el *id* del nodo asignador.
3. El nodo asignador después de haber concluido su temporizador (*timer1*) procede a detectar la cantidad de nodos vecinos presentes en la red validando con el número de tramas de respuesta recibidas. Si la cantidad de tramas que recibe el nodo es igual al número de nodos vecinos (vn) configurada, el nodo asignador procederá a ordenar la información de nivel de potencia de cada trama de mayor a menor; dado que se pueden detectar más nodos en la red, el nodo asignador únicamente admitirá la cantidad de nodos vecinos (vn) previamente configurada. El nodo vecino que posea una señal de potencia menor será el nuevo nodo asignador y el encargado de repetir automáticamente el proceso de asignación.
4. Después de procesar las tramas recibidas, el nodo asignador enviará las tramas de asignación con el identificador que será la posición x que tendrá cada nodo, además insertará en la trama una etiqueta que permitirá identificar al próximo nodo asignador. Se agrega la etiqueta con el valor de "Y" para asignar a un nodo como asignador y una etiqueta con el valor de "N" para identificar a un nodo como no asignador.

El proceso de direccionamiento en la cual se asigna los correspondientes identificadores cuando se tiene 3 nodos vecinos (vn) y una *Id* inicial de 0 se aprecia en la Figura 2.2.

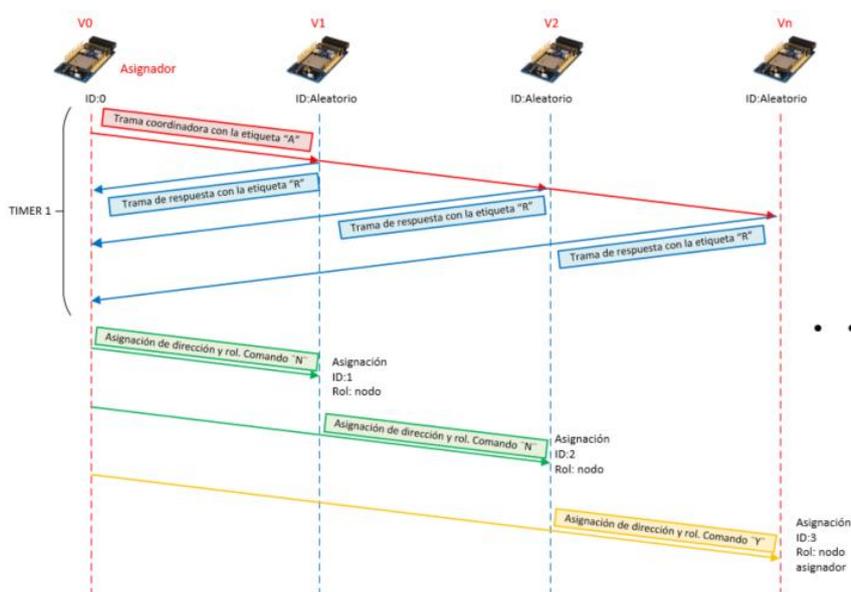


Figura 2.2. Diagrama del funcionamiento normal del algoritmo.

2.1.2.2. Funcionamiento en caso de enlaces fallidos

En situaciones donde existan fallas en la comunicación, producidas generalmente por circunstancias como: interferencias de señal, un medio de transmisión ocupado, un nodo ocupado o un nodo inactivo; estas situaciones ocasionan que los nodos no puedan recibir o enviar tramas correctamente. Debido a que no se usan acuses de recibo, el algoritmo para garantizar la correcta asignación de direcciones en su proceso debe realizar tareas adicionales que permitan a cada nodo obtener un identificador.

Los escenarios de enlace fallidos que contempla el algoritmo son: enlaces fallidos en la recepción de peticiones de asignación y enlaces fallidos durante el envío de identificadores.

2.1.2.2.1. *Enlaces fallidos en la recepción de peticiones de asignación*

Cabe recalcar que para poder probar este comportamiento en un escenario controlado, mediante programación, se simula el enlace fallido de un nodo vecino modificando su comportamiento para que no responda con peticiones de asignación al nodo asignador al menos una vez, para lo cual será necesario presionar el botón (SW2) ubicado en el módulo transceptor en el que se desee simular este comportamiento.

Para detectar los nodos que requieren un identificador, el nodo asignador envía una trama *broadcast* con la etiqueta "A" y activa un temporizador (*timer1*) que permite al nodo asignador contar y almacenar la información de las peticiones de asignación o tramas con la etiqueta "R", una vez finalizado el timer, el algoritmo compara la cantidad de tramas recibidas con la cantidad de nodos vecinos que espera encontrar en la zona de cobertura. Si el número de tramas recibidas es menor al número de nodos vecinos configurado, el nodo asignador retransmite la trama *broadcast* y repite el proceso previamente mencionado.

El algoritmo considera un número máximo de 3 retransmisiones de la trama *broadcast* de asignación, después de superar el número máximo de retransmisiones, si el número de tramas de respuesta sigue siendo menor al número de nodos vecinos, el algoritmo considera la presencia de nodos fallidos. Si no recibe ninguna trama de respuesta, finaliza su ejecución

En caso de recibir respuesta de todos los nodos vecinos el nodo asignador procederá a ordenar la información y enviará los identificadores con la posición (*id+x*) y el rol correspondiente a los nodos vecinos, asegurando de esta manera el direccionamiento.

El proceso de direccionamiento en caso de existir enlaces fallidos en la petición de asignación; configurado previamente con un identificador inicial igual a 0 y con 3 nodos vecinos (v_n) se presenta en la Figura 2.3.

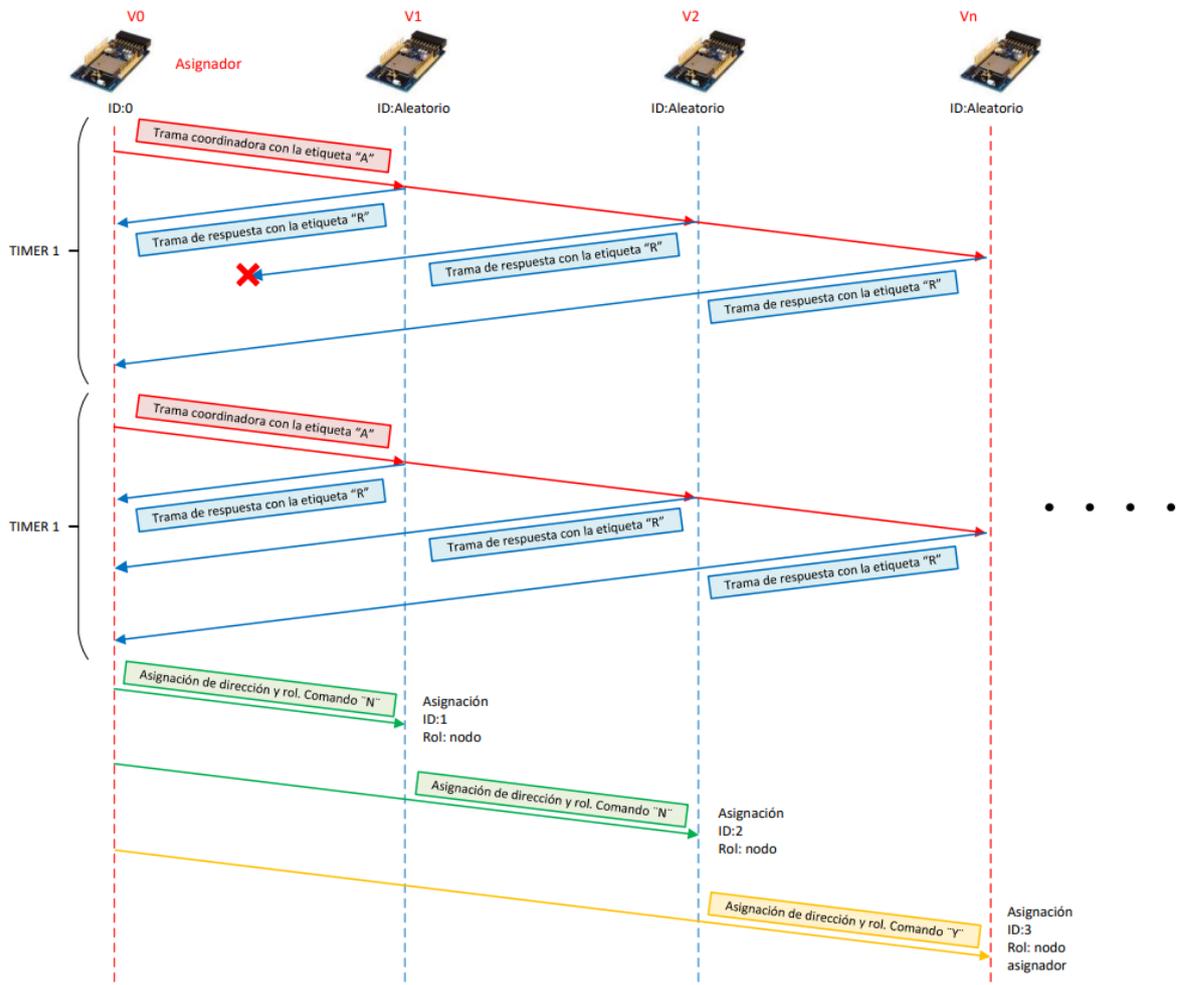


Figura 2.3. Diagrama de enlaces fallidos en la recepción de peticiones de asignación.

2.1.2.2.2. Enlaces fallidos durante el envío de identificadores.

Un enlace fallido puede producirse durante el envío de las tramas de asignación por parte de los nodos asignadores, ocasionando que las tramas no sean recibidas por los nodos frontera de la red o por los nodos cercanos al nodo asignador, en consecuencia, los nodos no podrían obtener un identificador de red.

El algoritmo para solventar este problema considera dos tipos de situaciones: cuando existe un nodo cercano que no recibe identificador y cuando el nodo de borde o frontera no recibe identificador.

- *Nodo de borde o frontera no recibe identificador.*

El fallo en la asignación de dirección al próximo nodo asignador puede ser crítico para la comunicación de toda la red, ya que no se podría repetir el proceso de asignación con los siguientes nodos de red. Para dar solución a este problema el nodo asignador después de haber enviado la última trama de asignación ejecuta un segundo temporizador (*timer2*) que esperará recibir la trama *broadcast* del siguiente nodo asignador en la red para poder confirmar de esta manera la correcta asignación del mismo, si no recibe el *broadcast* del siguiente nodo en la red procederá a reenviar mediante *unicast* la trama de asignación. Este método de comunicación es conocido como ACK implícito, en el cual no se usan acuses de recibo.

El proceso de direccionamiento con enlaces fallidos configurado previamente con un identificador inicial igual a 0 y con 2 nodos vecinos se presenta en la Figura 2.4.

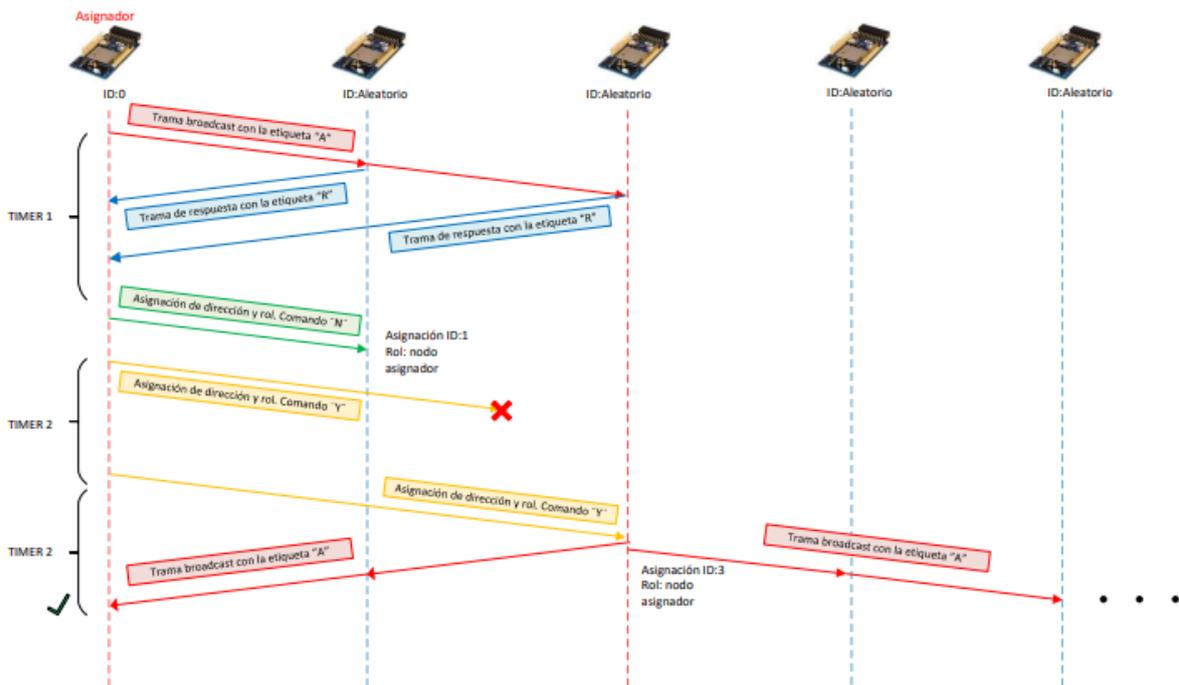


Figura 2.4. Diagrama en el que el nodo borde o frontera no recibe identificador.

- *Nodo cercano o intermedio no recibe identificador*

En caso de que un nodo vecino cercano no reciba su identificador, el nodo podrá responder al *broadcast* del siguiente nodo asignador en la red. En consecuencia, el siguiente nodo asignador (*vn*) receptorá una mayor cantidad de tramas de

respuesta después de enviar el *broadcast* de asignación. Para dar solución a este problema el nodo asignador enviará una trama con la etiqueta "E" al nodo asignador anterior para que vuelva a enviar las tramas de asignación con los identificadores correspondientes.

El proceso de direccionamiento con enlaces fallidos configurado previamente con un identificador inicial igual a 0 y con 2 nodos vecinos se presenta en la Figura 2.5.

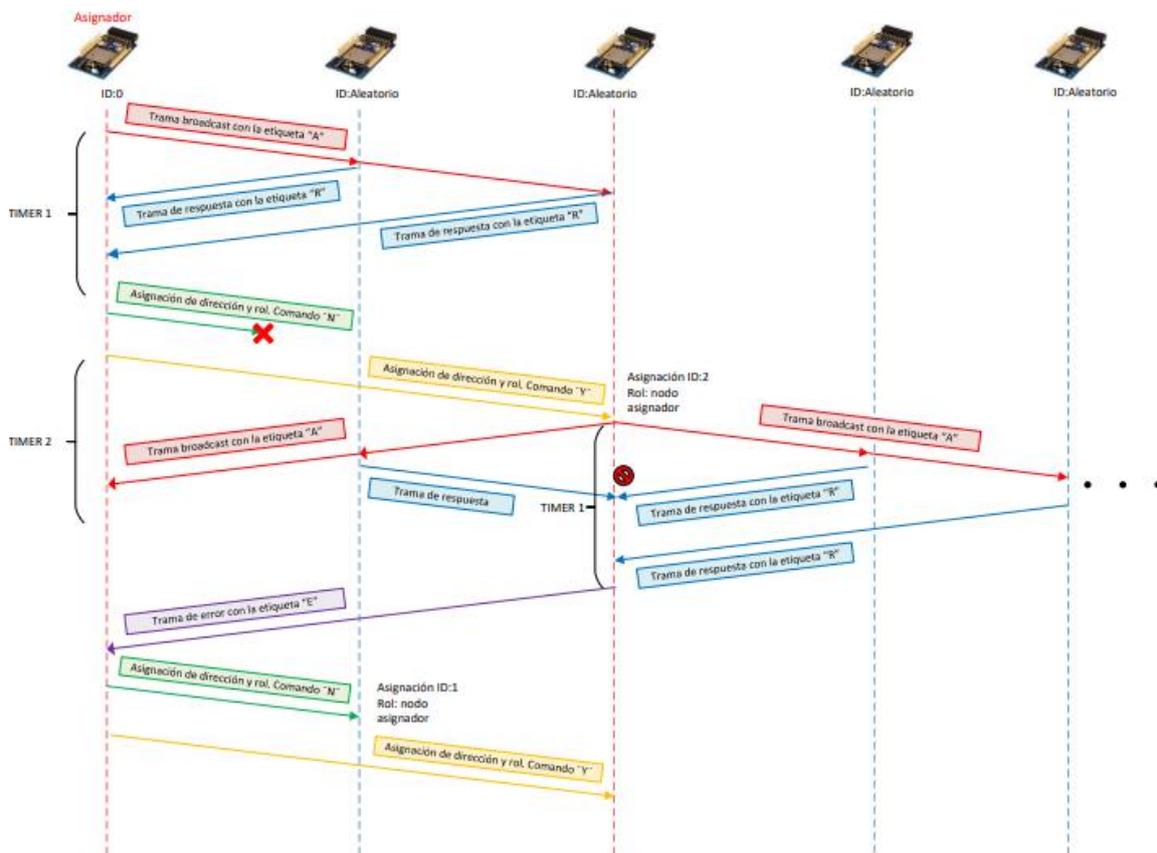


Figura 2.5. Diagrama en el que el nodo cercano o intermedio no recibe identificador.

2.1.2.3. Funcionamiento en caso de nodos fallidos

Si bien los nodos pueden haber sido probados previamente, puede suceder que estos presenten averías tales como: baterías defectuosas, fallos electrónicos o simplemente una mala configuración del nodo; el algoritmo con el fin de mantener una secuencialidad ascendente en los identificadores de red puede identificar qué direcciones no pueden ser asignadas.

El algoritmo para determinar qué identificador debe asignar hace uso de niveles de referencia que permiten obtener la posición del nodo en la estructura lineal. Un método

de ordenamiento simple no funciona debido a que la secuencialidad ascendente en la red se pierde dado que la posición del nodo se puede confundir.

Cabe recalcar que se deben encontrar niveles de referencia adecuados mediante una medición previa del nivel de potencia en la señal de los nodos vecinos, además, se deben considerar factores como la zona de cobertura de los nodos, la distancia entre los nodos, el nivel de energía (baterías) de los módulos, la temperatura del ambiente; factores que ocasionan una variabilidad en la medición del nivel de potencia.

El nodo asignador mediante un contador determina la cantidad de tramas de respuesta que recibió por parte de los nodos vecinos presentes en su zona de cobertura, si el número de tramas de respuesta es menor al número de nodos vecinos configurado, el nodo asignador procesa las tramas considerando la presencia de nodos fallidos. Para lo cual, el nodo asignador clasifica las señales recibidas y las ubica dentro de un nivel de referencia configurado. El nivel de referencia determina la posición del nodo vecino en la zona de cobertura del nodo asignador y la posición es usada como información en el envío de la trama de asignación para identificar correctamente a los nodos vecinos.

Ejemplo: Si se recibe una trama con un nivel de señal de -70 dBm y se tienen dos niveles de referencia con un umbral de -66 dBm. El nodo asignador considerará al nodo como el más alejado y por lo consiguiente enviará un identificador id más el nivel de referencia ($id+2$) . Ver Figura 2.6

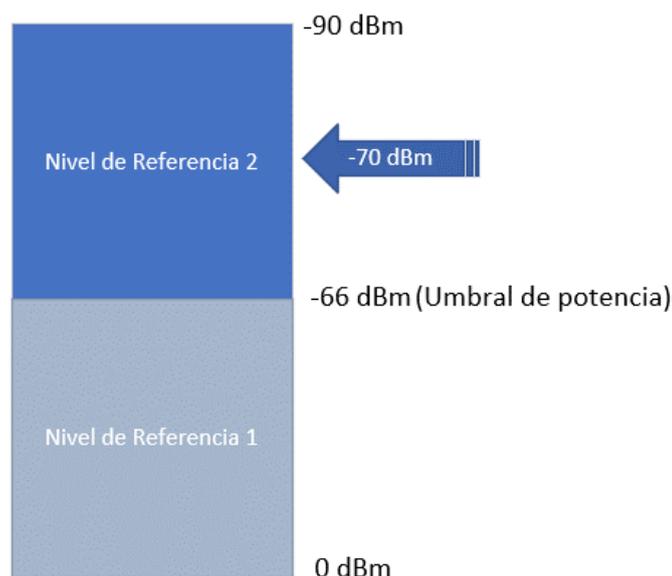


Figura 2.6. Clasificación de la señal según niveles de referencia.

El nodo asignador al finalizar la asignación de direcciones envía mediante una trama de prueba *broadcast* la información de los identificadores no asignados.

El proceso de direccionamiento con la presencia de un nodo averiado "V2"; configurado previamente con un identificador inicial igual a 0 y con 3 nodos vecinos "Vn" se presenta en la Figura 2.7.

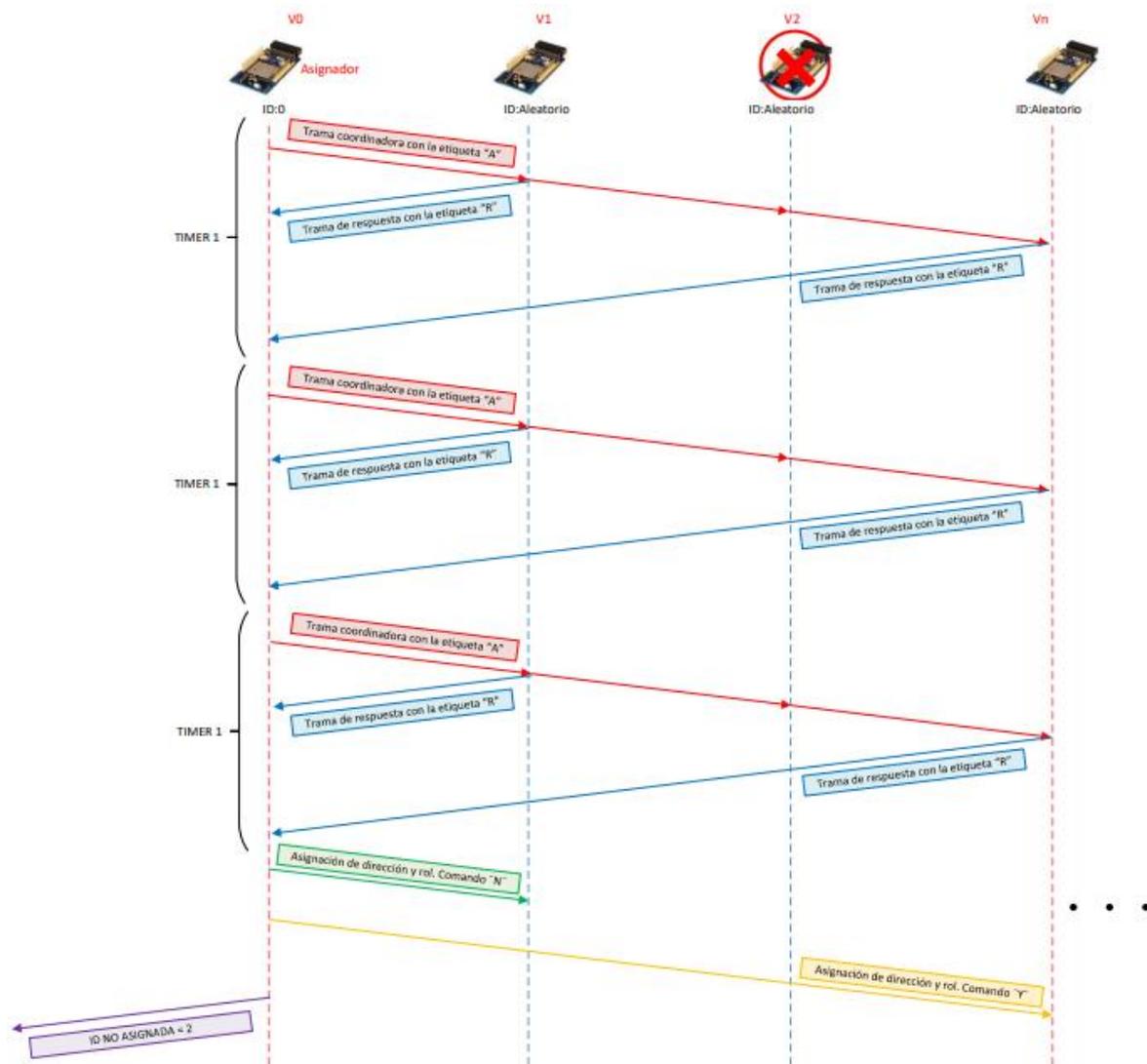


Figura 2.7. Diagrama del funcionamiento en caso de nodo fallido.

2.2. IMPLEMENTACIÓN DEL ALGORITMO

2.2.1. ANÁLISIS DE REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DEL ALGORITMO.

El algoritmo para su implementación requiere de *hardware*, *software* y una adecuada configuración del prototipo de pruebas. Requerimientos necesarios para codificar, implementar y probar el algoritmo de asignación de direcciones.

2.2.1.1. Requerimientos de Hardware para la implementación

Para la instalación y pruebas del algoritmo en los módulos ATZB-256RFR2-XPRO fue necesario contar con los siguientes elementos:

2.2.1.1.1. Herramienta programadora Atmel-ICE

Atmel-ICE nos permite la depuración y la programación de los módulos ATZB-256RFR2-XPRO a través de las interfaces *JTAG*, *SWD*, *PDI*, *TPI*, *aWire*, *SPI* y *debugWIRE*; siendo la interfaz *JTAG* la necesaria para cargar el código en el módulo ATZB. El dispositivo permite la depuración completa a nivel de fuente en *Atmel Studio*, es compatible con puntos de interrupción de *hardware* incorporados en el microcontrolador ATmega256RFR2 y permite hasta 128 puntos de interrupción de *software*. Requiere una alimentación mediante USB desde 1.62 V a 5.5V [34]. Observar la Figura 2.8.



Figura 2.8. Herramienta programadora Atmel-ICE [34].

2.2.1.1.2. Herramienta de captura de tramas o sniffer Zigbee CC2531 USB Dongle

La herramienta capturadora de tramas es un elemento esencial que nos permite observar la comunicación en la red de sensores y realizar pruebas del prototipo implementado. Para cumplir con el requerimiento se usó el dispositivo *Zigbee CC2531 USB Dongle*, el cual es un *sniffer* completamente operativo que proporciona una interfaz de PC para las

aplicaciones IEEE802.15.4 / *ZigBee*. El dispositivo posee un adaptador USB el cual se puede conectar directamente a la PC [35]. Observar la Figura 2.9.



Figura 2.9. Sniffer Zigbee CC2531 USB Dongle [35].

2.2.1.2. **Requerimientos de Software para la implementación**

Para la implementación del algoritmo de asignación de direcciones en una topología lineal se hizo uso del entorno de desarrollo de *Atmel Studio 7.0* junto con su extensión *Wireless Composer*. *Atmel Studio* es una plataforma de desarrollo necesaria para codificar el algoritmo usando lenguaje C/C++, mientras que *Wireless Composer* es una extensión de *Atmel* que agrega características del estándar IEEE 802.15.4. necesarias para implementar el algoritmo.

Además, se requirió de la herramienta *SmartRF™ Packet Sniffer*, la cual es una aplicación de *software* que permite mostrar y almacenar las tramas generadas por los nodos en la red, y que fueron capturadas por el dispositivo *Zigbee CC2531 USB Dongle*. La herramienta soporta distintas tecnologías entre ellos el estándar IEEE 802.15.4.

El entorno de desarrollo de *Atmel Studio 7.0*, así como la herramienta *SmartRF™ Packet Sniffer*, fueron instaladas en un computador con sistema operativo Windows 10.

2.2.1.3. **Rango de cobertura y ubicación de los módulos**

La especificación IEEE 802.15.4 contempla un rango de cobertura de hasta 60 metros en zonas con interferencia, por lo cual se probó en ambientes similares el módulo ATZB-256RFR2-XPRO para conocer su rango de cobertura, dando como resultado un alcance de 80 metros teniendo línea de vista, pero por efectos experimentales para poder realizar pruebas y mostrar los resultados necesarios del proyecto, en la implementación de nuestro prototipo se disminuyó la sensibilidad del receptor para trabajar con menores distancias.

Al reducir la sensibilidad de recepción, se obtuvo un alcance de 2 metros probados. Por lo que se decidió poner los nodos separados a una distancia de alrededor de 1 metro. Cabe recalcar que las distancias de separación no fueron necesariamente equidistantes, por la criticidad de usar el nivel de potencia para procesar las tramas recibidas, ya que esta se ve afectada también por la distancia entre los nodos. Se procuró mantener en la zona de cobertura de un nodo asignador al menos dos nodos vecinos, manteniendo siempre una topología lineal.

2.2.2. IMPLEMENTACIÓN DEL CÓDIGO

Para programar el microcontrolador ATmega256RFR2 que contiene el módulo ATZB-256RFR2-XPRO, se hizo uso del asistente de proyectos *IEEE 802.15.4 MAC Wireless Project Wizard* de la extensión *Wireless Composer* perteneciente a *Atmel Studio*; definida anteriormente en la sección 2.2.1.2.

La forma en que se creó y configuró el proyecto es muy similar a otros proyectos de titulación, por lo que de manera genérica se lo puede encontrar en la sección 2.4.1 de la tesis desarrollada por el Ing. Angel Erick Aguaguña Aconda [37].

Una vez configurado y establecido el proyecto, el asistente nos crea automáticamente los diferentes módulos y archivos necesarios para nuestra programación, como se muestra en la Figura 2.10.

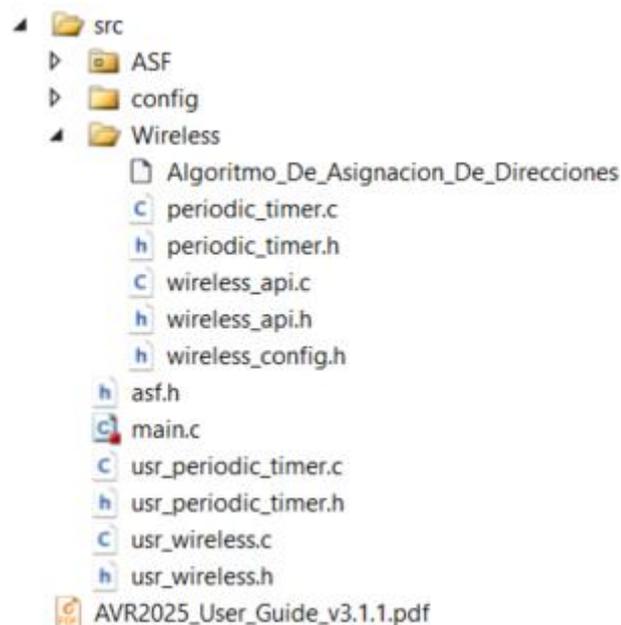


Figura 2.10. Estructura de archivos al crear un proyecto.

Los elementos que posee el proyecto son los siguientes:

- Archivo *main.c*: contiene la función principal del transceptor en donde se inicializan las funciones básicas del transceptor y se llaman continuamente a las funciones presentes en los archivos de usuario.
- Archivos de usuario: los ficheros *usr_wireless.c*, *usr_periodic_timer.c* y sus cabeceras *usr_wireless.h* , *usr_periodic_timer.h* permiten al usuario implementar la lógica de programación y manipular la aplicación según sus requerimientos. El fichero *usr_wireless.c* es donde se codifican las tareas que ejecuta cada transceptor.
- Carpeta “*Wireless*”: en esta carpeta se ubican los ficheros fuente que contienen las configuraciones de las propiedades inalámbricas, así como las propiedades de cada módulo complementario; creadas en función del asistente de proyectos.
 - Archivos inalámbricos: el fichero *wireless_api.c* y las cabeceras *wireless_api.h* y *wireless_config.h* contienen las funciones necesarias para configurar el transceptor, además provee de instrucciones básicas para la comunicación inalámbrica del estándar IEEE 802.15.4. La cabecera *wireless_config.h* contiene los parámetros de configuración para la comunicación de la red inalámbrica como la frecuencia, canal de transmisión, dirección de la red PAN, etc.
 - Archivos de módulos complementarios: son archivos generados de acuerdo con la cantidad de funcionalidades extra que se desee implementar en el transceptor. Dado que se necesitó el uso de temporizadores se creó automáticamente el fichero *periodic_timer.c* y su cabecera *periodic_timer.h*, los cuales contienen las respectivas funcionalidades y configuraciones del temporizador.

Mediante código se definen los procesos que llevan a cabo los nodos para ejecutar el algoritmo de asignación de direcciones. Procesos que se definieron anteriormente en la sección 2.1 y que se pueden observar de manera desglosada en el diagrama de flujo de las Figuras 2.11 y 2.12.

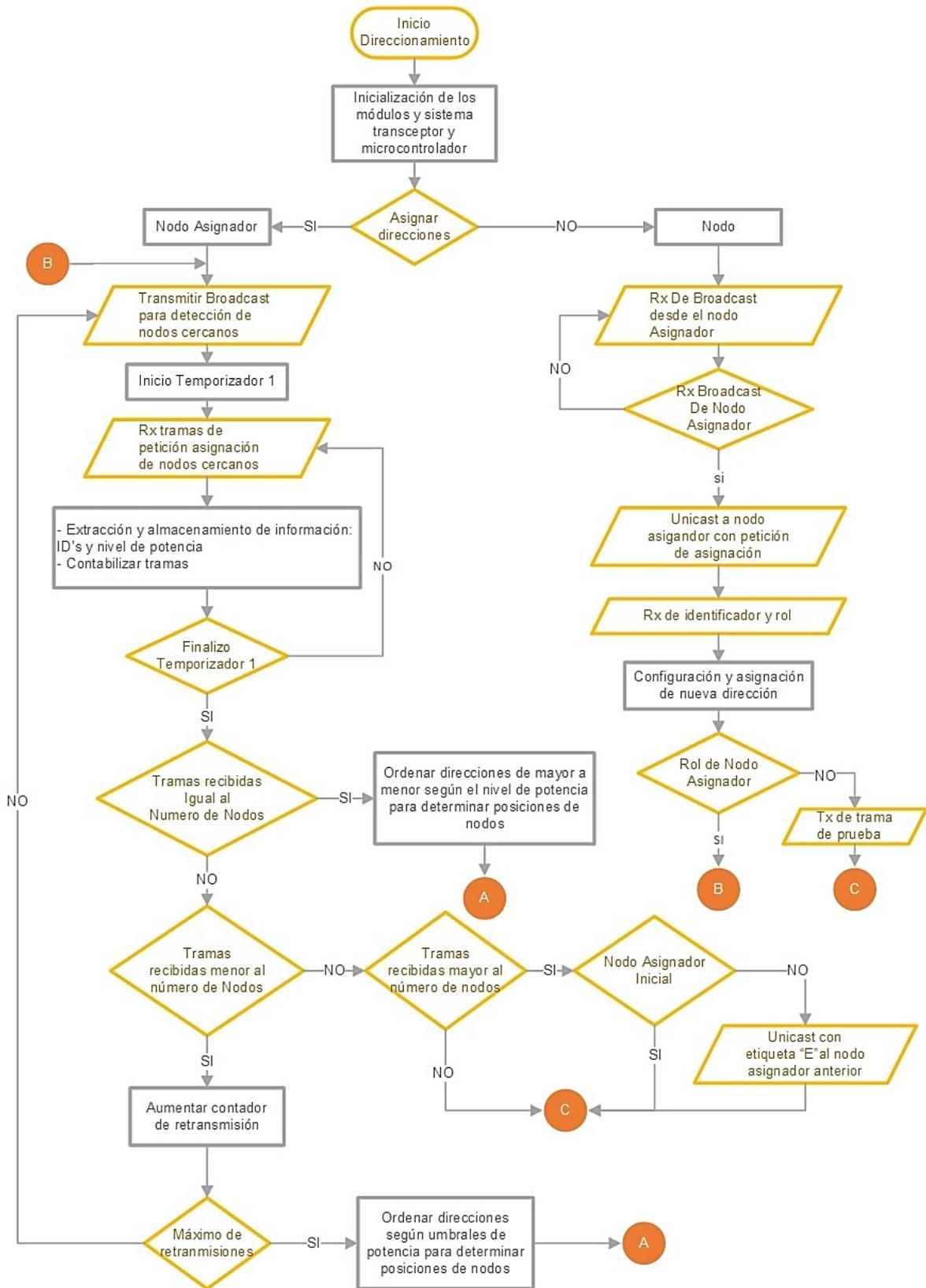


Figura 2.11. Diagrama de flujo para el proceso de asignación de direcciones (Parte 1).

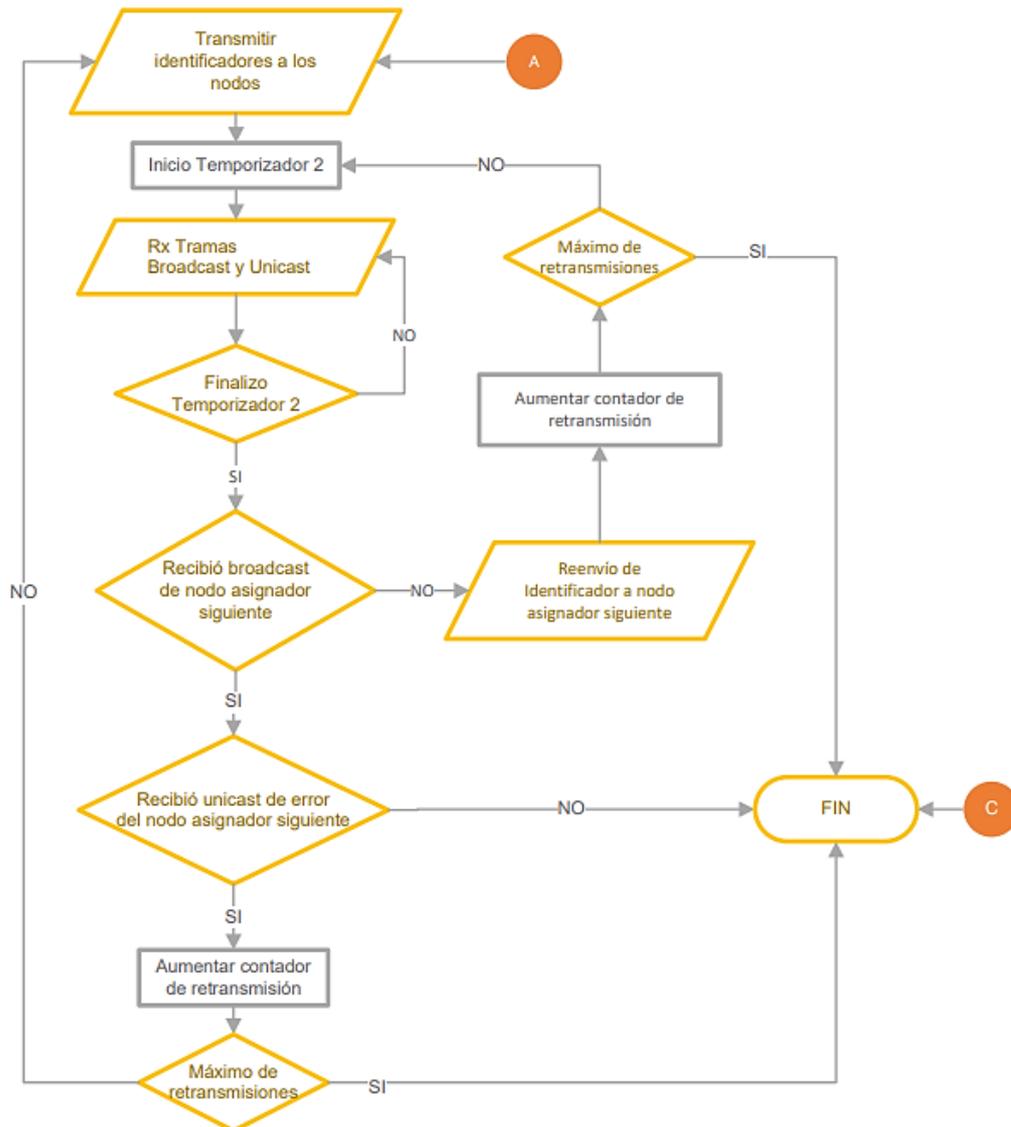


Figura 2.12. Diagrama de flujo para el proceso de asignación de direcciones (Parte 2).

2.2.2.1. Inicialización y configuración de componentes

Para proceder con la inicialización de componentes del microcontrolador, tales como *LEDs*, botones, comunicación serial, etc., y para configurar los parámetros y tareas iniciales que el nodo transceptor realizará al encenderse, es necesario su parametrización en los archivos destinados para su configuración.

Dentro del archivo *conf_board.h* ubicado en la carpeta “config” se encuentra la configuración para hacer uso de los *LEDs* y del botón SW2. El botón SW2 nos permite iniciar los procesos del algoritmo. Ver Código 2.1.

```

/*Archivo de configuracion de botones y leds*/
#ifndef CONF_BOARD_H
#define CONF_BOARD_H

//Se inicializa los leds con sus respectivos puertos I/O
#define LED1R IOPORT_CREATE_PIN(PORTD, 6)
#define LED2G IOPORT_CREATE_PIN(PORTG, 2)
#define LED3Y IOPORT_CREATE_PIN(PORTE, 2)

//Se inicializan el boton SW2 con su respectivo puerto I/O
#define GPIO_PUSH_BUTTON_0 IOPORT_CREATE_PIN(PORTE, 0)
#define GPIO_PUSH_BUTTON_ON_BOARD GPIO_PUSH_BUTTON_0

//Se definen los estados de los leds
#define APAGAR_LED(led_gpio)      ioport_set_pin_level(led_gpio, 1)
#define ENCENDER_LED(led_gpio)   ioport_set_pin_level(led_gpio, 0)
#define CONMUTAR_LED(led_gpio)   ioport_toggle_pin_level(led_gpio)

#endif // CONF_BOARD_H

```

Código 2.1 Configuración de LEDs y botón SW2.

Para configurar los parámetros de la red inalámbrica, es decir los parámetros de la PIB como: PAN ID, modo de operación CSMA no ranurado, petición de acuses de recibo, canal, velocidad de transmisión, entre otros, se usa el archivo *wireless_config.h*. Estos parámetros se crean por defecto al configurar el proyecto con el asistente de proyectos. Aquí se ubica el identificador de la red PAN (PAN_ID). Ver Código 2.2

```

//Configuración del transceptor
#define TRANSMITTER_ENABLED

//Identificadores PAN ID
#define DEFAULT_PAN_ID      0xCAFE
#define SRC_PAN_ID         0xCAFE
#define DST_PAN_ID         0xCAFE

//Configuración del canal de transmision
#define CHANNEL_TRANSMIT_RECEIVE  20
//Configuración de la velocidad de transmision
#define CHANNEL_PAGE_TRANSMIT_RECEIVE  0

//Configuración de acuses de recibo, repetición de tramas, modo CSMA
#define ACK_REQ              0
#define FRAME_RETRY         0
#define CSMA_MODE           CSMA_UNSLOTTED

```

Código 2.2 Configuración de parámetros de la red PAN.

Los parámetros del transceptor definidos anteriormente en el Código 2.2 son usados en el archivo *wireless_api.c*, el cual fue generado automáticamente por el asistente de proyectos y contiene todas las funciones necesarias para inicializar y operar el transceptor. Aquí se ubica la configuración de direcciones aleatorias iniciales de todos los nodos. Cabe recalcar que existe una posibilidad extremadamente baja del 0.0001%

(1/65535) de que estos identificadores se repitan, en caso de que se produzca conflicto por la generación de un *id* duplicado o reservado simplemente se puede reiniciar el dispositivo. Ver Código 2.3

```
//CCPU_ENDIAN_TO_LE16: Conversion de constantes desde CPU endian a little endian
uint16_t pan_id = CCPU_ENDIAN_TO_LE16(SRC_PAN_ID);
uint8_t channel = CHANNEL_TRANSMIT_RECEIVE;
uint8_t channel_page = CHANNEL_PAGE_TRANSMIT_RECEIVE;

uint16_t src_addr = 1 + (rand() % (65535-1)); // Se usa la funcion rand() para
la creacion de una variable aleatoria que puede tomar valores desde 0x0001 hasta
0xFFFFE

src_addr = CCPU_ENDIAN_TO_LE16(src_addr); //Se configura una direccion aleatoria
al inicio

/* Configurar la direccion inicial por defecto */
tal_pib_set(macShortAddress, (pib_value_t *)&src_addr);

/* Configuracion PAN ID. */
tal_pib_set(macPANId, (pib_value_t *)&pan_id);

/* Configuracion del canal. */
tal_pib_set(phyCurrentChannel, (pib_value_t *)&channel);

/* Configuración de la velocidad de transmision*/
tal_pib_set(phyCurrentPage, (pib_value_t *)&channel_page);
```

Código 2.3. Configuración de direcciones aleatorias iniciales.

El archivo *main.c* generado automáticamente llama a las funciones y parámetros definidos previamente, en la función principal *int main (void)* de la aplicación, se inicializa el módulo inalámbrico y los demás módulos de la aplicación. Además, contiene un bucle con la función principal *WirelessTask()*, la cual llama continuamente a funciones que poseen tareas relacionadas con las capas PAL y TAL; y la función de aplicación en la cual se programa el algoritmo. Cabe recalcar que este tipo de funciones son proporcionadas por la librería AVR2025 MAC y generadas automáticamente. Ver Código 2.4.

```

int main(void)
{
    /* Inicializa el módulo inalámbrico - wireless_api */
    wireless_init();

    /*Se inicializa el boton, leds y temporizadores */
    modules_init();

    while (1)
    {
        //Funcion que contiene las tareas de las capas TAL y PAL
        WirelessTask();
    }
}

```

Código 2.4. Función principal *main*.

En el archivo *usr_wireless.c* se inicializan las variables globales que serán usadas a lo largo del código, las librerías y también se definen las constantes o parámetros necesarios para el funcionamiento del algoritmo; como el tipo de nodo, numero de nodos esperados en la zona de cobertura de un nodo asignador, el máximo de retransmisiones en caso de enlaces fallidos, los umbrales de potencia para el caso de nodos fallidos, etc. En el Código 2.5 se definen las librerías y constantes.

```

//Librerias
/*-----*/
#include "usr_wireless.h" // Archivo que contiene el campo de la trama y estructuras
que son parte de la logica de aplicacion
#include "wireless_config.h" // Archivo que contiene informacion de la PIB (Canal,PAN
ID,Velocidad de transmision,etc)
#include "periodic_timer.h" // Archivo que contiene informacion sobre los
temporizadores 1 y 2
//Constantes
/*-----*/
#define NUMERO_DE_NODOS 2 /**< Define la cantidad de nodos esperados dentro de la
zona de cobertura*/
#define MAX_RETRANSMISION 3 /**< Define cuantos intentos permitidos de retransmision
de asignacion se va a tener*/

```

Código 2.5 Definición de librerías y constantes.

En el código 2.6 se definen las variables globales.

```

/*-----*/
//      DECLARACION DE VARIABLES GLOBALES
/*-----*/
//VARIABLES BOOLEANAS.
bool asignador = false; //Variable booleana para identificar el tipo de nodo y sus funciones a realizar
bool transmision = false; //Variable booleana para activar o permitir la transmision de tramas
bool procesamiento = false; //Variable booleana para activar el procesamiento de las tramas y su envio
bool simulacion_enlace_fallido = false; //Variable booleana que permite activar la simulacion de enlace
fallido mediante la pulsacion del boton solo una vez
bool simulacion_enlace_fallido2 = false; //Variable booleana que permite activar la simulacion de enlace
fallido mediante la pulsacion del boton dos veces.
bool direccion_asignada = false; //Variable booleana que permite identificar si el nodo cuenta con una
direccion asignada

```

```

bool id_duplicada=false; //Variable booleana que permite detectar si se encontraron dos o mas nodos dentro
de un umbral en el caso de nodos fallidos
bool error_asignacion_intermedio=false; //Variable booleana que permite activar de nuevo la asignacion de
direcciones
bool asignacion_nodo_asignador = false; //Variable booleana que permite asignar de nuevo la direccion al
nodo asignador en caso de no recibir la direccion
bool asignacion_finalizada = false; //Variable booleana que permite realizar los procesos siguientes a la
asignacion de direcciones en el caso de enlaces fallidos
//CONTADORES
uint8_t retransmision=1; //Contador de numero de intentos de transmision de la trama de asignacion
uint8_t cont_trx_asignador=0; //Contador de las tramas recibidas con peticion de asignacion
uint8_t cont_ttx_asignador=0; //Contador de tramas enviadas de asignacion
uint8_t cont_presion_boton=0; //Contador para definir el tipo de nodo y sus funciones, por defecto nodo no
asignador
//VARIABLES
char comando_a_transmitir='0'; //Variable que permite identificar que tipo de comando se va a transmitir
char payload[4]; //Variable que permite armar el payload de envio
uint16_t direccion_destino; //Variable que permite identificar la direccion destino para enviar una trama
uint16_t direccion_a_asignar; //Variable que permite identificar la nueva direccion asignada al nodo
uint16_t direccion_nodo_asignador; //Variable que permite identificar la direccion del nodo asignador
uint16_t posicion_asignacion; //Variable que me permite armar el payload y enviar la posicion
uint16_t PAN_ID = CCPU_ENDIAN_TO_LE16(DEFAULT_PAN_ID); // Identificador de la red PAN
Trama trama_tx; //Variable que permite obtener la trama enviada y manipularla.
Trama trama_rx; //Variable que permite obtener la trama recibida y manipularla.
//VECTORES
Info_Direccionamiento tabla_direccionamiento[NUMERO_DE_NODOS+2]; //Vector que contiene informacion de
direccionamiento(direccion,potencia,posicion)
int8_t umbrales_de_potencia[]={0,-50,-63}; // contiene los limites de cada umbral de potencia para poder
realizar la comparacion
uint8_t n_umbrales = (uint8_t)sizeof(umbrales_de_potencia); //Contiene el numero de umbrales que contiene el
vector de umbrales de potencia
uint8_t num_nodos[NUMERO_DE_NODOS]; //Vector va a contener el numero nodos dentro de un umbral determinado
//TEST
bool mostrar_direcciones_en_led=false; //Variable booleana temporal que me permite ver la direccion asignada
en binario en los leds correspondientes
Medicion_De_Potencia tabla_calculo_potencia[NUMERO_DE_NODOS+2]; //Tabla que sirve para realizar calculos de
potencia promedio y clasificar la tabla de direccionamiento calculada
Info_Direccionamiento tabla_direccionamiento_acumulada[(NUMERO_DE_NODOS+2)*MAX_RETRANSMISION]; //Tabla que
contiene informacion de direccionamiento de las tramas de respuesta de asignacion
uint8_t cont_registros_direccionamiento=0; //Contador que me permite aumentar los registros en la tabla de
direccionamiento acumulada.

```

Código 2.6. Definición de Variables Globales

Para obtener y manipular una trama también es necesario crear una estructura que contenga como variables los campos del formato de la trama IEEE 802.15.4, campos como la longitud de trama, control de secuencia, direcciones origen y destino, etc. Esta estructura se la define en el archivo *usr_wireless.h* para poder ser llamado por las funciones implementadas en el archivo *usr_wireless.c*. Ver Código 2.7.

```

typedef struct{
    uint8_t longitud_trama; /**< Encabezado PHY (Longitud de la trama) */
    uint16_t FCF; /**< Campo de control de la trama*/
    uint8_t numero_secuencia; /**< Numero de secuencia de la trama*/
    uint16_t PAN_destino; /**<Identificador Pan Destino */
    uint16_t dir_destino; /**<Identificador de direccion de destino */
    uint16_t dir_origen; /**<Identificador de direccion de origen*/
    char payload[MAX_PAYLOAD]; /**<Payload de la trama */
    uint16_t FCS; /**<CRC16 campo de la trama*/
}Trama;

```

Código 2.7. Estructura que contiene los campos de la trama IEEE 802.15.4.

De la misma manera, para poder guardar información de direccionamiento, se crea una estructura que permite simular una especie de tabla de direccionamiento, la cual contiene información de origen de las tramas recibidas, su nivel de potencia y la posición de los nodos vecinos. Ver Código 2.8

```
typedef struct{
    int8_t potencia;//Potencia extraida de la trama de recepcion
    uint16_t direccion;//Direccion de origen de la trama
    uint8_t posicion_calculada;//Numero de saltos hasta llegar al nodo
}Info_Direccionamiento;
```

Código 2.8. Estructura que contiene la información de direccionamiento.

Finalmente, también se crea un nuevo tipo de dato *enum* para poder distinguir el tipo de procesamiento que se llevará a cabo por los nodos. Ver Código 2.9

```
}typedef enum{
    NINGUNO,
    ASIGNACION_NORMAL,
    ASIGNACION_NODO_FALLIDO,
    SETEAR_DIRECCION,
    DETECCION_NODOS_FALLIDOS,
    ENVIO_DE_INFORMACION_DE_NODOS_FALLIDOS,
    ENVIO_RESUMEN_POTENCIA_ASIGNACION,
}SHORTENUM Tipo_de_proceso;
```

Código 2.9. Definición de objeto tipo *enum* para distinguir los tipos de procesos.

2.2.2.2. Código de asignación de direcciones

En el archivo *usr_wireless.c* es donde principalmente se programa el algoritmo de asignación de direcciones, este archivo contiene las funciones necesarias para programar tanto las tareas de un nodo asignador como las de un nodo simple. Estas funciones nos permiten realizar procesos para transmitir tramas, recibir tramas, procesar tramas enviadas, procesar interrupciones de los temporizadores e implementar la lógica de aplicación del algoritmo de direccionamiento.

Para cada nodo se codifica sus respectivas tareas las cuales en su conjunto nos permiten construir el algoritmo de asignación de direcciones, dicha tareas se explicarán a continuación en pequeños fragmentos de código. Cabe recalcar que el código completo se lo encuentra en el ANEXO I.

- **Tarea de transmisión de broadcast asignación del nodo asignador**

Para determinar los nodos vecinos que requieren un identificador es necesario enviar una trama *broadcast* con la etiqueta “A” la cual nos permite sondear los nodos presentes en el área de cobertura de un nodo asignador. Esta tarea se

puede ejecutar presionando tres veces el botón SW2 del módulo transceptor o cuando un nodo haya sido elegido como nuevo asignador. Para el envío de la trama *broadcast* de asignación se considera el número máximo de retransmisiones. Ver Código 2.10

```
switch(comando_a_transmitir){//Segun el comando a transmitir configurado
    en el procesamiento y en la funcion de respuesta
    case 'A':
        if(retransmision<=MAX_RETRANSMISION){ //Si el contador de
            retransmision es menor al numero maximo de retransmisiones
            transmit_sample_frame((uint8_t*)"A",1,BROADCAST); //
            Transmitir broadcast de asignacion con la letra "A"
            transmission=false;//Desactivar la transmission
        }
        break;
```

Código 2.10. Envío de *broadcast* de asignación para sondeo de nodos

Además, para enviar tramas se usa la función de la capa TAL llamada `transmit_sample_frame((uint8_t*)"datos", tamaño, dirección destino)`, para ello se debe especificar los datos que va a llevar el *payload*, su tamaño en bytes y la dirección destino.

Después de enviar la trama *broadcast*, en la función de tramas enviadas `usr_frame_transmitted_cb()`, se activa el timer 1 que permite dar un lapso de tiempo de espera por las tramas de respuesta que contienen la petición de asignación de los nodos vecinos. Ver Código 2.11

```
if (asignador)
{
    memset(&trama_tx,0,sizeof(trama_tx));//Vaciar la estructura en la que se va a
    copiar la trama transmitida
    memcpy(&trama_tx,frame->mpdu,sizeof(trama_tx));//Copiar la trama enviada en
    la estructura
    char comando_enviado = trama_tx.payload[0];//Extraer el comando enviado del
    payload de la trama enviada
    switch (comando_enviado){//Segun el comando enviado en la trama transmitida
        case 'A':
            start_timer1();//Inicializar el timer que espera por la respuesta de
            la trama de asignacion
        break;
```

Código 2.11. Inicio del timer 1 después de enviar el *broadcast*.

- **Tarea de extracción, almacenamiento y contabilización de tramas de respuesta del nodo asignador**

Después de inicializar el timer1, en la función de tramas recibidas `usr_frame_received_cb()` el nodo asignador puede contabilizar las tramas de

respuesta que recibe de los nodos vecinos. El nodo asignador también extrae el nivel de potencia de las tramas y su dirección origen para almacenarlos en la tabla de direccionamiento. El nodo asignador solo procesa las tramas de respuesta con la etiqueta "R" y si es que estas pertenecen a la misma red PAN, como se puede observar en el Código 2.12.

```

if(trama_rx.payload[0]=='R' && trama_rx.PAN_destino==PAN_ID){
    //Extraccion de potencia y conversion en dbm
    int8_t rssi_base_val, valor_ed;//Variables que permiten extraer el nivel
    de potencia de la trama recibida
    rssi_base_val = tal_get_rssi_base_val();//Obtener el valor base de RSSI
    del nodo que permite convertir el valor ED a dbm
    uint8_t phy_frame_len = frame->mpdu[0];//Extraer la longitud de la trama
    para ubicar la posicion ED dentro de la misma
    valor_ed = frame->mpdu[phy_frame_len + LQI_LEN + ED_VAL_LEN] +
    rssi_base_val;//Mapear el valor de potencia ED a dbm

    tabla_direccionamiento
    [cont_trx_asignador].direccion=trama_rx.dir_origen;//Guardar la
    direccion de origen de la trama en la estructura
    tabla_direccionamiento[cont_trx_asignador].potencia=valor_ed;//Guardar la
    potencia en dbm de la trama
    cont_trx_asignador++; //Contabilizacion de tramas
}

```

Código 2.12. Extracción, almacenamiento y contabilización de tramas de respuesta del nodo asignador.

- **Tareas de envío de respuesta al mensaje broadcast.**

Los nodos vecinos que reciben el *broadcast* del nodo asignador y que no poseen una dirección asignada, pueden responder con una trama *unicast* con la etiqueta "R". Además, si está activa la simulación de enlaces fallidos el nodo no envía la trama de respuesta, como se puede apreciar en el Código 2.13.

```

...
if (!asignador){
    //Si el nodo intermedio recibe el comando con la letra "A", si la trama
    pertenece a la misma red PAN y si aun no tiene
    //una direccion asignada
    if(trama_rx.payload[0] == 'A' && trama_rx.PAN_destino==PAN_ID &&
    trama_rx.dir_destino == BROADCAST && !direccion_asignada && !
    simulacion_enlace_fallido){
        ENCENDER_LED(LED1R);
        ENCENDER_LED(LED2G);
        ENCENDER_LED(LED3Y);

        comando_a_transmitir='R'; //Establecer el comando proximo a
        transmitir en "R"
        direccion_destino=trama_rx.dir_origen; //Establecer la direccion de
        destino de la trama proxima a transmitir
        transmision=true; //Activar la transmision
    }
}

```

Código 2.13. Envío de respuesta a mensaje *broadcast* de asignación.

- **Tareas de procesamiento de tramas recibidas por el nodo asignador**

Después de que finaliza el timer 1 se ejecuta una interrupción la cual llama a la función *usr_app_timer_cb1()*. En esta función se deciden las tareas a realizar en base al número de tramas de respuesta recibidas, se ordenan las potencias de mayor a menor y se clasifican las potencias según su umbral de potencia. Se realiza este procesamiento con el objetivo de asignar identificadores según los tres escenarios de asignación: asignación normal, asignación con enlaces fallidos y asignación con nodos fallidos.

En el código 2.14. se observa la implementación del ordenamiento de mayor a menor de las potencias en la tabla de direccionamiento y el cálculo de posiciones de los nodos vecinos; las posiciones podrán identificar a los nodos vecinos y son las que se enviarán en la tarea de transmisión de identificadores a los nodos. El ordenamiento se realiza en caso de una asignación normal o en caso de una asignación con enlaces fallidos.

```
if (cont_trx_asignador == NUMERO_DE_NODOS)
{
    int8_t m,n;//Variables contadores usadas para los lazos
    Info_Direccionamiento temp; //Variable temporal usada para ordenar los
    registros de la tabla de direccionamiento
    //ORDENAMIENTO DE LA POTENCIA SEGUN EL METODO DE INSERCCION
    for(m=1;m<NUMERO_DE_NODOS;m++){
        temp=tabla_direccionamiento[m];
        n=m-1;
        //Lazo que ordena de mayor a menor
        while(n>=0 && tabla_direccionamiento[n].potencia < temp.potencia){
            tabla_direccionamiento[n+1]=tabla_direccionamiento[n];
            n=n-1;
        }
        tabla_direccionamiento[n+1]=temp;
    }
    //Lazo que asigna la posicion calculada al vector de estructuras ordenado
    for(int8_t k=0;k<cont_trx_asignador;k++){
        tabla_direccionamiento[k].posicion_calculada=k+1;
    }
    procesamiento=true;//Activar el procesamiento
    id_proceso=ASIGNACION_NORMAL;//Establecer el tipo de procesamiento proximo a
    realizar en ASIGNACION NORMAL
    retransmision=0;//Reiniciar el contador de retransmision
}
```

Código 2.14. Ordenamiento de la tabla de direccionamiento según el nivel de potencia.

En el caso de nodos fallidos el ordenamiento se lo hace usando umbrales de potencia, estos umbrales de potencia permiten definir la posición del nodo vecino fallido y reservar su dirección. Cabe recalcar que los umbrales de potencia se los define previamente y dependerán del número de nodos vecinos cercanos. Ver Código 2.15

```
//Lazo que permite realizar la asignacion del nivel de potencia segun los
//umbrales de potencia definidos
for(uint8_t l=0;l<cont_trx_asignador;l++){//Lazo para recorrer el vector de
//estructuras
    for(uint8_t k=0;k<n_umbrales;k++){
        //Condicional de comparacion entre los umbrales definidos
        if(tabla_direccionamiento[l].potencia<umbrales_de_potencia[k] &&
        tabla_direccionamiento[l].potencia>=umbrales_de_potencia[k+1]){
            tabla_direccionamiento[l].posicion_calculada = k+1;//Se asigna
            //el nivel del umbral
            break;
        }
    }
}
```

Código 2.15. Ordenamiento según umbrales de potencia para caso de nodo fallido.

- **Tareas de transmisión de identificadores a los nodos**

Después de que finaliza el timer 1 y los identificadores de los nodos son calculados, se procede a enviar las tramas. Este proceso se realiza de manera ordenada según la información contenida en la tabla de direccionamiento la cual permite conocer la dirección de destino de las tramas a enviar y determinar tanto el rol como la posición del nodo a configurar la dirección.

La última trama se enviará con la etiqueta “Y” y será la que identificará al próximo nodo asignador, para los demás nodos se enviará la etiqueta “N”; la implementación de esta lógica de envío se puede ver en el Código 2.16.

```
if (id_proceso==ASIGNACION_NORMAL || id_proceso==ASIGNACION_NODO_FALLIDO)
{
    //Si la trama que se envia es la ultima
    if (cont_ttx_asignador==cont_trx_asignador-1)
    {
        comando_a_transmitir='Y';//Establecer el proximo comando a enviar
        //en la letra "Y" en la ultima trama de envio
    }else{
        comando_a_transmitir='N'; //Establecer el proximo comando a
        //enviar en la letra "N" para las primeras tramas de envio
    }
}
```

```

//Si hubo una asignacion fallida a un nodo intermedio al terminar la
//asignacion y el comando a transmitir es la Y
if (error_asignacion_intermedio && comando_a_transmitir=='Y')
{
    //En la repeticion del proceso, el nodo asignador cercano ya no
    //tendra direccion aleatoria
    direccion_destino=tal_pib.ShortAddress + cont_trx_asignador;//
    //Establecer la direccion destino del siguiente nodo asignador
}
//Caso contrario es la primera vez que el nodo asigna las direcciones
else
{
    direccion_destino=tabla_direccionamiento
    [cont_ttx_asignador].direccion;//Obtener la direccion de la
    //tabla de direccionamiento para ser usada en transmision
}
posicion_asignacion=tabla_direccionamiento
[cont_ttx_asignador].posicion_calculada; //Obtener la posicion de
//la tabla de direccionamiento para ser usada en transmision
transmision=true;//Activar la transmision
procesamiento=false;//Desactivar el procesamiento
}
}

```

Código 2.16. Lógica de envío de los identificadores.

Después de enviar la última trama de asignación, el nodo asignador activa el timer2 el cual permite esperar un lapso de tiempo para recibir el *broadcast* o recibir la trama *unicast* de error con la etiqueta “E” del siguiente nodo asignador. Este método nos permite determinar si las direcciones fueron enviadas correctamente o hubo un enlace fallido cuando se enviaron los identificadores. En caso de que el nodo asignador detecte cualquiera de estas dos tramas detendrá el timer2, si se recibe una trama con etiqueta “E” se repetirá el proceso de asignación. Ver Código 2.17

```

else if (trama_rx.payload[0]=='E' && trama_rx.PAN_destino==PAN_ID &&
trama_rx.dir_destino == tal_pib.ShortAddress && asignacion_finalizada) //
//Caso de error de asignacion de nodo intermedio. Recibe unicast
{
    //Si el proceso de error de asignacion de nodo intermedio aun no ha sido
    //activado
    if (!error_asignacion_intermedio)
    {
        cont_ttx_asignador=0;//Reiniciar el contador de tramas de asignacion
        //para poder enviar de nuevo las tramas
        id_proceso = ASIGNACION_NORMAL;//Establecer el tipo de procesamiento
        //a asignacion normal para repetir el proceso de asignacion
        procesamiento = true;//Activar el procesamiento
    }
}

```

```

        error_asignacion_intermedio = true;//Activar el error de asignacion
        de nodo intermedio para enviar las tramas correspondientes
        stop_timer2();//Detener el timer 2 que espera detectar la recepcion
        de la trama que contiene el comando "E"
        asignacion_nodo_asignador = false;//Establecer la asignacion del nodo
        asignador en falso para indicar que este nodo aun no ha sido
        asignado la direccion
    }
}
//Si el nodo asignador recibe la trama broadcast de asignacion, pertenece a
la misma PAN y la asignacion de direcciones ha finalizado
else if (trama_rx.payload[0]=='A' && trama_rx.PAN_destino==PAN_ID &&
asignacion_finalizada)
{
    asignacion_nodo_asignador = true;//Establecer la asignacion del nodo
    asignador en true para indicar que este nodo ya ha sido asignado la
    direccion
    stop_timer2();//Detener el timer 2 que espera por la trama broadcast que
    contiene el comando "A"
}
}

```

Código 2.17. Confirmación de envío de tramas de asignación para prevenir enlaces fallidos.

En caso de no recibir el *broadcast* del siguiente nodo asignador, una vez finalizado el timer2 se ejecutará la interrupción que llamará a la función *usr_app_timer_cb2()*. Esta función permite acceder al último registro de la tabla de direccionamiento que contiene la posición del siguiente nodo asignador y permite reenviar la trama al siguiente nodo asignador. Ver Código 2.18

```

//Si la asignacion de direccion al nodo asignador no fue realizada
if (!asignacion_nodo_asignador)
{
    //Si el contador de retransmision de la trama de asignacion es menor al
    numero maximo de retransmisiones configurado
    if (retransmision<MAX_RETRANSMISION)
    {
        retransmision++;//Aumentar el contador de retransmision
        cont_ttx_asignador = cont_trx_asignador-1;//Establecer el contador de
        transmission que se utiliza para acceder al ultimo registro de la tabla
        de direccionamiento
        id_proceso = ASIGNACION_NORMAL;//Establecer el tipo de procesamiento
        proximo a realizar en ASIGNACION NORMAL
        procesamiento=true;//Activar el procesamiento
    }
}
}

```

Código 2.18. Reenvío de identificador al nodo asignador siguiente.

Cabe recalcar que se permite un máximo de 3 retransmisiones en cualquier caso de reenvío de tramas por enlaces fallidos.

- **Tarea de recepción de tramas de asignación y configuración de nueva dirección**

Los nodos que aún no poseen una dirección asignada recibirán la trama enviada por parte del nodo asignador con información de direccionamiento. El nodo vecino asumirá el rol de nuevo asignador si en el primer byte, del *payload* de la trama recibida, lee la etiqueta “Y”; para así repetir el proceso de asignación y conformar la topología lineal. Si el nodo lee la etiqueta “N” solo configura automáticamente su nueva dirección.

El nodo para configurar su nueva dirección, en el segundo byte del *payload* de la trama recibida, leerá su posición respecto al nodo asignador. Esta posición le permite autoasignarse su identificador en base a la dirección de su nodo asignador, es decir una identificación “*Id* + *x*”, donde “*Id*” es la dirección de su nodo asignador y “*x*” es el identificador de posición recibida por parte del nodo asignador, su suma determinará la dirección asignada. Ver Código 2.19.

```
if(trama_rx.payload[0] == 'Y' && trama_rx.PAN_destino==PAN_ID && !
direccion_asignada && !simulacion_enlace_fallido2){
    char dir[2] //Variable que permite guardar el identificador en ascii
    uint8_t identificador; //Variable que permite guardar el identificador en
        entero
    direccion_nodo_asignador=trama_rx.dir_origen; //Obtener la direccion del
        nodo asignador
    dir[0]=trama_rx.payload[1]; //Obtener el identificador/posicion del payload
    dir[1]='\0'; //Cerrar la cadena
    identificador=atoi(dir); //Convertir de caracter a numero entero sin signo
    //Calculo de la direccion a configurar segun el identificador recibido
    direccion_a_asignar = direccion_nodo_asignador + (uint16_t)identificador;
    asignador=true; //Establecer al nodo como nuevo asignador
    id_proceso=SETEAR_DIRECCION; //Establecer el tipo de procesamiento a SETEAR
        DIRECCION para configurar la direccion del nodo
    procesamiento=true; //Activar el procesamiento
}
if(trama_rx.payload[0] == 'N' && trama_rx.PAN_destino==PAN_ID && !
direccion_asignada && !simulacion_enlace_fallido2){
    char dir[2]; //Variable que permite guardar el identificador en ascii
    uint8_t identificador; //Variable que permite guardar el identificador en
        entero
    direccion_nodo_asignador=trama_rx.dir_origen; //Obtener la direccion del
        nodo asignador
    dir[0]=CCPU_ENDIAN_TO_LE16(trama_rx.payload[1]); //Obtener el identificador
        posicion del payload
```

```

dir[1]='\0';//Cerrar la cadena
identificador=atoi(dir);//Convertir de caracter a numero entero sin signo
//Calculo de la direccion a configurar segun el identificador recibido
direccion_a_asignar = direccion_nodo_asignador + identificador;
id_proceso=SETEAR_DIRECCION;//Establecer el tipo de procesamiento a SETEAR
    DIRECCION para configurar la direccion del nodo
procesamiento=true;//Activar el procesamiento
}

```

Código 2.19. Recepción de tramas de asignación.

Para configurar la dirección corta se hace uso de la función *tal_pib_set(macShortAddress, (pib_value_t *) "dirección")* proporcionada por la capa TAL, como se puede ver el Código 2.20.

```

if (id_proceso==SETEAR_DIRECCION){//Se activa cuando recibe las tramas
con el comando N
    if (MAC_SUCCESS==tal_pib_set(macShortAddress, (pib_value_t *)
&direccion_a_asignar))//Si se logra asignar
    {
        comando_a_transmitir='T';           //Proximo comando a enviar es
        la letra "T"
        transmision=true;                   //Activar la transmision
        id_proceso=NINGUNO;                 //Indicar el siguiente
        procesamiento en ninguno
        direccion_asignada=true;           //Cambiar estado, direccion
        ya asignada
        procesamiento=false;               //Desactivar el procesamiento
    }else{
        //Si no es posible asignar la direccion porque la capa TAL esta
        ocupada repetir el proceso
        id_proceso=SETEAR_DIRECCION;       //Realizar el proceso de
        setear direccion
        direccion_asignada=false;           //Cambiar estado, direccion
        aun no asignada
        procesamiento=true;                 //Activar el procesamiento
    }
}
}

```

Código 2.20. Configuración de dirección.

2.3. EVALUACIÓN DE LOS RETARDOS

Para proponer un modelo matemático y evaluar teóricamente el tiempo que demorará el algoritmo de asignación en enviar las direcciones en una topología lineal a gran escala que opera con el estándar IEEE 802.15.4, es necesario identificar las variables que influyen en el tiempo de asignación en un escenario “ideal” sin errores.

Con estas variables se busca proyectar de manera teórica el retardo del algoritmo de asignación de direcciones y desarrollar una fórmula matemática que permita obtener el valor del retardo en el caso de tener cientos de nodos y distintos saltos de red.

Para el análisis se tomó como base los tiempos para una red con 3 nodos, donde el nodo asignador V0 asigna direcciones a dos nodos vecinos. Ver Figura 2.13.

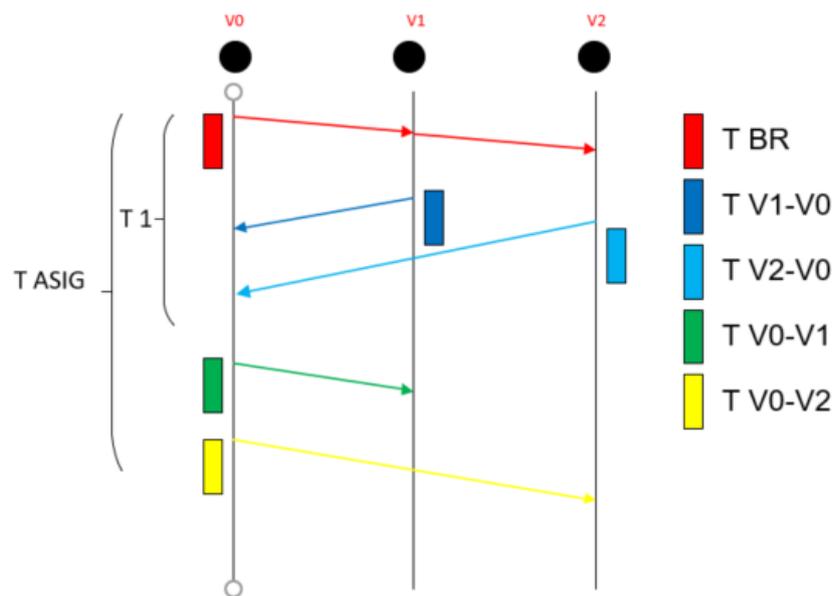


Figura 2.13. Tiempos que intervienen en la asignación de direcciones

T BR	Tiempo de envío de la trama <i>broadcast</i> de detección.
T V1-V0	Tiempo de envío de respuesta por parte del nodo V1 al nodo V0
T V2-V0	Tiempo de envío de respuesta por parte del nodo V2 al nodo V0
T V0-V1	Tiempo de envío de identificador por parte del nodo V0 al nodo V1
T V0-V2	Tiempo de envío de identificador por parte del nodo V0 al nodo V2
T 1	Tiempo de espera para recibir respuestas de los nodos V1 y V2
T ASIG	Tiempo total de asignación para un salto de red.

El tiempo total de asignación es igual a la suma del tiempo T1 más el tiempo T V0-V1 y T V0-V2 en un caso de dos nodos vecinos. Proyectando para n nodos vecinos y considerando que el tiempo de comunicación entre el nodo V0 con el nodo V1 y el tiempo de comunicación entre el nodo V0 con el nodo V2 son los mismos, en un escenario ideal, se encuentra la siguiente Ecuación 2.1.

$$T_{ASIG} = T_1 + n T_n$$

Ecuación 2.1. Cálculo del tiempo de asignación para un solo salto considerando T1 (Temporizador 1).

Donde Tn representa el tiempo de comunicación entre un par de nodos.

$$T_n = T_{V0-V1} = T_{V0-V2}$$

El tiempo T1 es igual a la suma del tiempo T_{BR} más el tiempo T_{V1-V0} y T_{V2-V0} en un caso de dos nodos vecinos. De igual manera se puede considerar a los tiempos de comunicación entre los nodos vecinos V1 y V2 con el nodo asignador V0 como los mismos. Obteniendo T1 de la siguiente manera:

$$T_1 = T_{BR} + n T_n$$

Donde:

$$T_{BR} = T_n$$

Obteniendo finalmente el tiempo de asignación de la siguiente manera:

$$T_{ASIG} = T_n + n T_n + n T_n$$

Simplificando obtenemos el cálculo del tiempo de asignación para un solo salto representado en la Ecuación 2.2.

$$T_{ASIG} = T_n (1 + 2n)$$

Ecuación 2.2. Cálculo del tiempo de asignación un solo salto.

En una red multisalto, el primer salto va a tener 3 nodos en total requiriendo 1 tiempo de asignación. Con el segundo salto se va a tener 5 nodos en total requiriendo 2 tiempos de asignación, sumando otro salto se va a tener 7 nodos en total requiriendo 3 tiempos de asignación y así sucesivamente. Mediante este análisis se proyecta con la Ecuación 2.3. el tiempo de asignación total.

$$T_{ASIGTOTAL} = \left[\frac{x}{n} \right] T_{ASIG}$$

Ecuación 2.3. Cálculo del tiempo de asignación total.

Donde en la Ecuación 2.2 “x” es un número impar que representa la cantidad de nodos presentes en la red multisalto de topología lineal, y “n” el número de nodos vecinos dentro de la zona de cobertura de un nodo asignador. Cabe recalcar que se toma el entero inferior para obtener el tiempo de asignación que necesitaría un salto de red.

Para calcular el tiempo de comunicación entre nodos TN, se toma en cuenta el tiempo involucrado en el proceso que le toma al nodo V1 transmitir la trama de respuesta al nodo V0. Ver Ecuación 2.4.

$$T_n = T_{back_off} + T_{Ta} + T_p + T_{CCA} + T_{Trama} + T_{procesamiento}$$

Ecuación 2.4. Tiempo involucrado en la comunicación de dos nodos.

Donde:

- T_{Ta} representa el tiempo para pasar de estado de recepción a transmisión
- T_{PDDU} es el tiempo de transmisión de la trama.
- $T_{procesamiento}$ es el tiempo de ejecución del algoritmo.
- $T_{back-off}$ es el tiempo de espera aleatorio generado por el mecanismo CSMA/CA.
- T_{CCA} tiempo para censar el canal.
- T_p es el tiempo de propagación

El tiempo de propagación se asume como cero porque las distancias entre los nodos no se consideran significativas, así como el tiempo de pasar de estado de recepción a estado de transmisión. Por lo que para la comunicación entre un par de nodos el tiempo se calcula con la Ecuación 2.5.

$$T_n = T_{back_off} + T_{CCA} + T_{Trama} + T_{procesamiento}$$

Ecuación 2.5. Cálculo del tiempo de comunicación.[37]

El tiempo de *backoff* se calcula mediante la Ecuación 2.6 provista por el mecanismo CSMA/CA.

Ecuación 2.6. Cálculo del tiempo de back off. [37]

En donde la unidad de periodo de *backoff* es un valor constante de 320 microsegundos, el valor de la variable n depende del número de intentos a transmitir. [37]

$$T_{back_off} = (2^n - 1) * 320 \text{ us} = 320 \text{ us}$$

El tiempo de la trama se calcula mediante la Ecuación 2.7, en donde V_{tx} es la velocidad utilizada para la transmisión entre los nodos definida por el estándar en 250 Kbps.

$$T_{trama} = \frac{(Tamaño\ MPDU + SHR + PHR) * 8}{V_{tx}}$$

Ecuación 2.7. Cálculo del tiempo de la trama. [37]

$$T_{trama} = \frac{(13 + 5 + 1) * 8}{250\ Kbps} = 0.608\ ms$$

El tiempo de procesamiento del algoritmo es de 2.5 ms obtenidos mediante instrucciones de código previstas por el lenguaje de programación C/C++ [37].

Finalmente el tiempo teórico T_n es:

$$T_n = 320\ us + 128\ us + 608\ us + 2500us = 3556\ us$$

El tiempo teórico T_n nos permite proyectar como ejemplo el tiempo de asignación para 101 nodos con 2 nodos vecinos, usando la Ecuación 2.2 y 2.3. Ver Figura 2.14

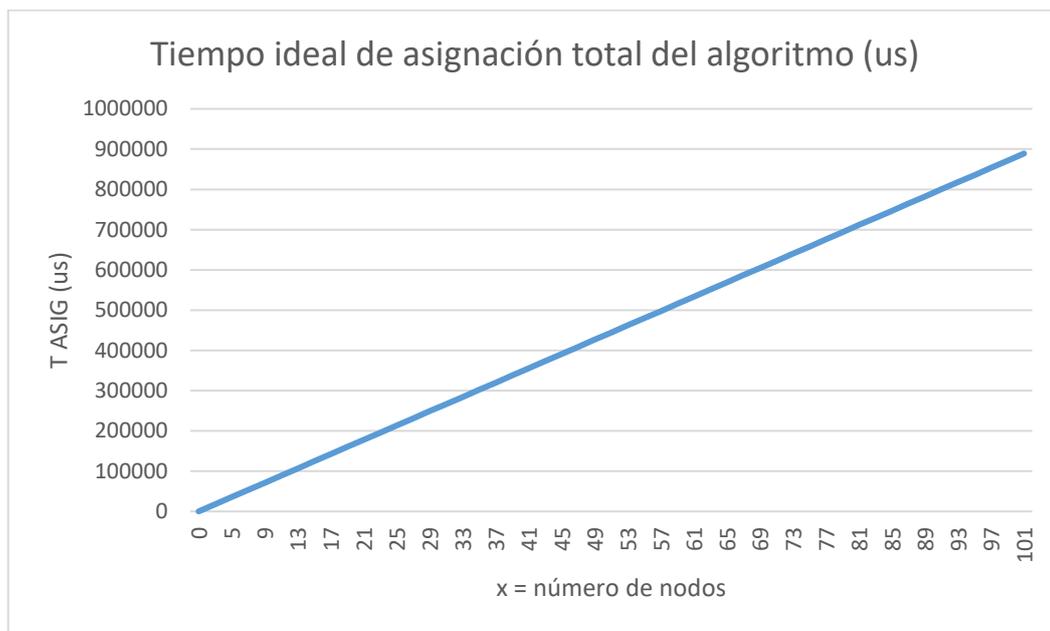


Figura 2.14. Proyección del algoritmo de direccionamiento para cientos de nodos

En la Figura 2.14 se muestra que se necesitaría alrededor de 0.9 segundos para asignar direcciones a 101 nodos en una topología lineal multisalto sin ningún tipo de error en la comunicación.

3. RESULTADOS Y DISCUSIÓN

En esta sección se describen los resultados de la ejecución del algoritmo en nuestro prototipo de red, así como la topología de red lineal creada con sus diferentes componentes.

3.1. CONFORMACIÓN DE LA TOPOLOGIA LINEAL

3.1.1 COMPONENTES USADOS EN LAS PRUEBAS.

Para formar nuestro prototipo de red disponemos de los siguientes componentes: 7 módulos ATZB-256RFR2-XPRO de *Atmel*, un computador con Windows 10, un *sniffer* IEEE 802.15.4 con su herramienta visualizadora de tramas *SmartRF™ Packet Sniffer* y 7 pares de pilas AAA para energizar a los módulos.

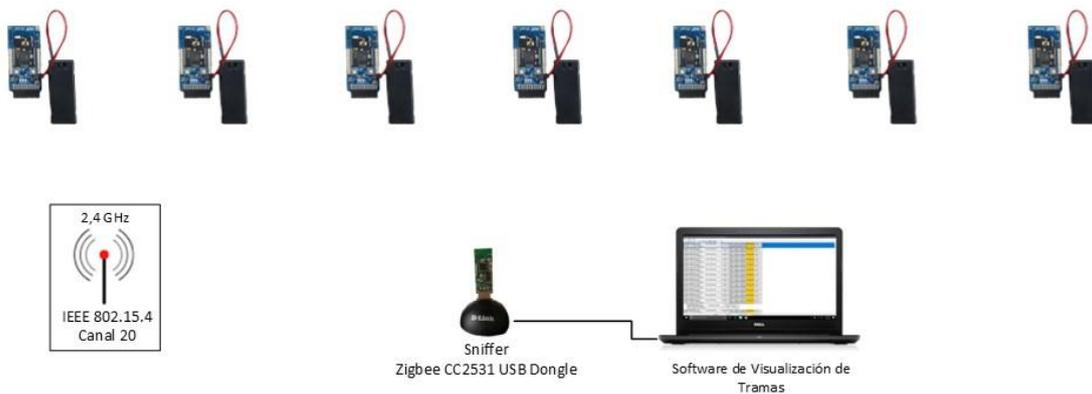


Figura 3.1 Componentes del prototipo de prueba.

3.1.2 IMPLEMENTACIÓN DEL PROTOTIPO DE RED INALÁMBRICA DE PRUEBA.

Para probar el algoritmo de asignación de direcciones se creó la topología lineal de prueba que puede ser observada en la Figura 3.2.

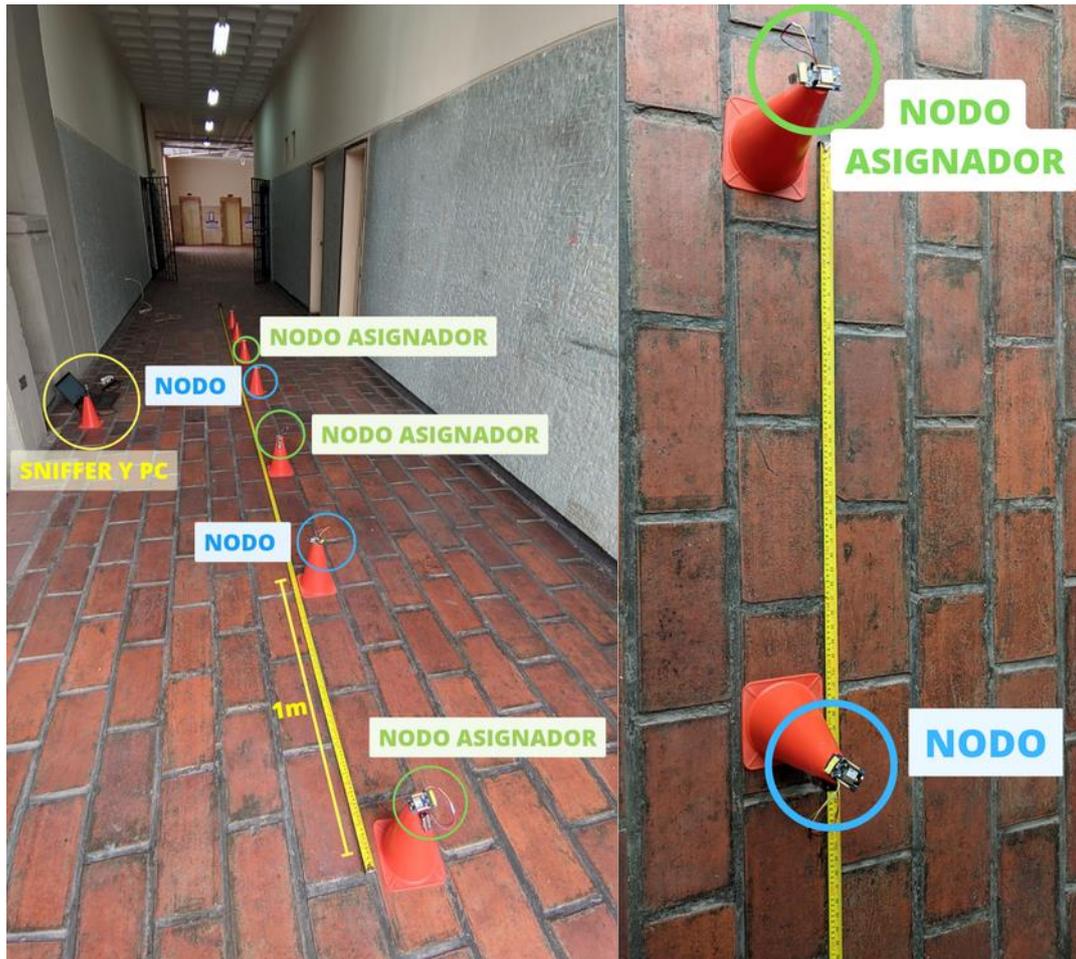


Figura 3.2. Fotografías de pruebas en un pasillo de la Escuela Politécnica Nacional.

Los nodos sensores se ubicaron de tal manera que dentro de la zona de cobertura de un futuro nodo asignador se encuentren dos nodos vecinos. Con distancias de separación de entre 1 y 1.5 metros, necesarios para obtener una red multisalto. Los nodos sensores poseen línea de vista, con una altura de 22 cm. Se procuró tener un ambiente libre de interferencias, para obtener los escenarios ideales de asignación.

Para comprobar que todos los nodos en la red obtengan su respectiva dirección, se colocó un *sniffer* en el centro de la red inalámbrica para poder observar en la herramienta visualizadora de tramas la dirección de origen de las tramas generadas en el proceso de asignación de direcciones.

Para grabar el algoritmo de asignación de direcciones, realizado y descrito en la sección de implementación de código (2.2.2), se usa la interfaz JTAG que posee el nodo ATZB-256RFR2-XPRO y la herramienta *ATMEL-ICE* que permite grabar el algoritmo en el nodo. Dado que solo es un programa para todos los nodos, el algoritmo es grabado y ejecutado en cada uno de los módulos.

3.2. PRUEBAS DE MEDICIÓN DE POTENCIA

El objetivo de la prueba es determinar los umbrales de potencia que permitan a los nodos operar correctamente en caso de nodos fallidos presentes en la red. Adicionalmente, debido a que la lectura de los niveles de potencia de las señales recibidas puede tener valores muy cercanos entre sí, esta prueba también permite determinar la cantidad de nodos vecinos (V_n) que se pueden colocar en la zona de cobertura de un nodo asignador, para prevenir situaciones en las que no se pueda clasificar adecuadamente la información de potencia de las tramas recibidas por la cercanía de los niveles de señal.

Para esta prueba se usaron 2 nodos, los cuales fueron separados a distancias de 10, 20, 30, 40, 50 y 60 metros. Cabe recalcar que los nodos fueron elevados a una altura de 50 cm para poder obtener una adecuada línea de vista. En cada separación se fueron tomando los valores en dBm de las señales recibidas con la ayuda de un *sniffer* de red.

Los resultados obtenidos tomando 3 medidas, se muestran en la Tabla 3.1 a continuación:

Tabla 3.1. Medición de Potencia

Separación de los nodos (m)	Medición 1 (dBm)	Medición 2 (dBm)	Medición 3 (dBm)	Promedio (dBm)
10	-68	-69	-69	-68,67
20	-77	-78	-78	-77,67
30	-81	-75	-83	-79,67
40	-83	-80	-82	-81,67
50	-85	-82	-81	-82,67
60	-83	-82	-85	-83,33

Analizando los resultados de la potencia promedio de la Tabla 3.1, podemos observar que, a distancias mayores, es decir a partir de los 40 metros, los niveles de señal recibida presentan valores muy cercanos entre sí. Como resultado no podemos tener un margen aceptable para clasificar la señal dentro de un nivel de referencia adecuado. En la Figura 3.1. podemos observar que a medida que la distancia aumenta, el margen de los niveles de referencia disminuye, lo que ocasiona que solo se pueda tener un umbral de potencia para poder discernir y clasificar adecuadamente las señales recibidas. Al tener solo un

umbral de potencia se limita la cantidad de nodos vecinos que un nodo asignador puede identificar en su zona de cobertura en caso de un nodo fallido, a solo “dos nodos”.

Cabe recalcar que el umbral de potencia cambia en diferentes ambientes y en distintas distancias, lo que dificulta discernir adecuadamente un nodo caído sin calibrar previamente el prototipo de red.

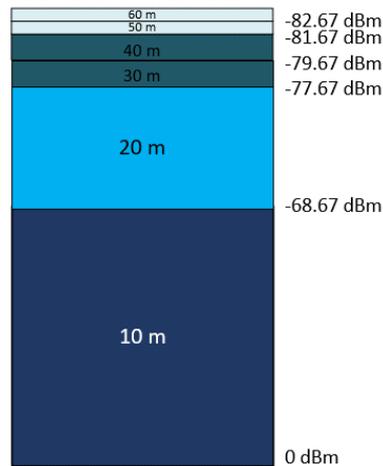


Figura 3.3. Medición de niveles de señal recibida según la distancia

3.3. PRUEBAS REALIZADAS DEL ALGORITMO DE ASIGNACIÓN DE DIRECCIONES

Para cada caso de prueba se procuró en la topología lineal tener al menos dos nodos vecinos dentro del área de cobertura de un nodo asignador, conformando de esta manera una red con topología lineal multisalto como se puede ver en la Figura 3.4.

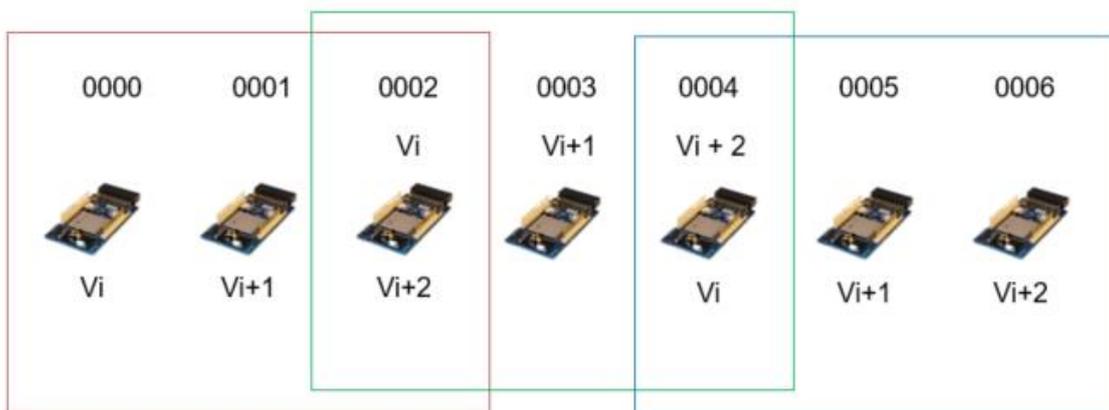


Figura 3.4. Topología lineal de prueba

3.3.1. PRUEBA 1

Asignación sin errores de comunicación.

El objetivo de la prueba es comprobar el correcto funcionamiento del proceso de direccionamiento por parte de los nodos asignadores de red a los nodos que forman parte de la red inalámbrica con topología lineal.

Una vez que los nodos transeptores se encuentren separados a una distancia adecuada y conformen una red con topología lineal, se procede a presionar tres veces el botón SW2 del transeptor en el nodo inicial, para dar comienzo al proceso de asignación de direcciones. El nodo asignador inicial configura su dirección de origen con 0000 y procede a determinar la dirección de 2 nodos vecinos cercanos.

En la Figura 3.4 podemos observar la información de envío y recepción de datos del proceso de asignación mediante un *sniffer* de red, el cual nos permite capturar tramas que operan con el estándar IEEE 802.15.4.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-69	OK
RX 2	+2439 =2439	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x6991	R	-54	OK
RX 3	+4597 =7036	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x24DC	R	-56	OK
RX 4	+27531 =34567	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x6991	0x0000	N 1	-68	OK
RX 5	+2071 =36638	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x24DC	0x0000	Y 2	-68	OK
RX 6	+3518 =40156	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0002	A	-56	OK
RX 7	+2743 =42899	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x41BA	R	-55	OK
RX 8	+3014 =45913	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x3EFC	R	-70	OK
RX 9	+27509 =73422	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x41BA	0x0002	N 1	-56	OK
RX 10	+1751 =75173	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x3EFC	0x0002	Y 2	-56	OK
RX 11	+2590 =77763	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0004	A	-71	OK
RX 12	+1161 =78924	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0004	0x3472	R	-75	OK
RX 13	+3637 =82561	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0004	0x0FE1	R	-73	OK

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+29728		Type Sec Pnd Ack.req PAN_compr					N		
14	=112289	13	DATA 0 0 0 1	0x02	0xCAFE	0x3472	0x0004	1	-71	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+3030		Type Sec Pnd Ack.req PAN_compr					Y		
15	=115319	13	DATA 0 0 0 1	0x03	0xCAFE	0x0FE1	0x0004	2	-70	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+1629		Type Sec Pnd Ack.req PAN_compr					A		
16	=116948	12	DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0006	A	-73	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+33634		Type Sec Pnd Ack.req PAN_compr					A		
17	=150582	12	DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0006	A	-74	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+35622		Type Sec Pnd Ack.req PAN_compr					A		
18	=186204	12	DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0006	A	-75	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+16532811		Type Sec Pnd Ack.req PAN_compr					T		
19	=16719015	12	DATA 0 0 0 1	0x01	0xCAFE	0x0003	0x0003	T	-56	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+234687		Type Sec Pnd Ack.req PAN_compr					T		
20	=16953702	12	DATA 0 0 0 1	0x01	0xCAFE	0x0005	0x0005	T	-78	OK
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+151761		Type Sec Pnd Ack.req PAN_compr					T		
21	=17105463	12	DATA 0 0 0 1	0x01	0xCAFE	0x0001	0x0001	T	-54	OK

Figura 3.5. Captura de tramas prueba 1

La trama con número de secuencia 1 observado al inicio de la trama de la Figura 3.5, es generada por el nodo con dirección origen 0000 el cual inicia la transmisión con un *broadcast* inicial, esta trama es captada por dos nodos vecinos los cuales responden con una trama de respuesta capturadas en las tramas 2 y 3. Seguidamente, el nodo asignador envía las tramas 4 y 5 para asignar los identificadores correspondientes. El nodo vecino de borde con dirección aleatoria inicial, al recibir la trama de asignación, se asigna la dirección 0002 y vuelve a repetir el proceso; que puede ser observado desde la trama 6 hasta la 10. De la misma manera el nuevo nodo asignador con dirección 0004 vuelve a repetir el proceso visto desde la trama 11 hasta la 15. Por último, el nodo asignador frontera con dirección 0x0006 al no recibir más respuestas termina el proceso de asignación enviando 3 tramas *broadcast* como puede verse desde la trama 16.

La correcta asignación de los identificadores 0001, 0003 y 0005 de los nodos vecinos intermedios puede ser vista en la dirección de origen de las tramas de prueba 19, 20 y 21.

Sin contar con las tramas de prueba, lo explicado anteriormente se resume de manera gráfica en la Figura 3.6.

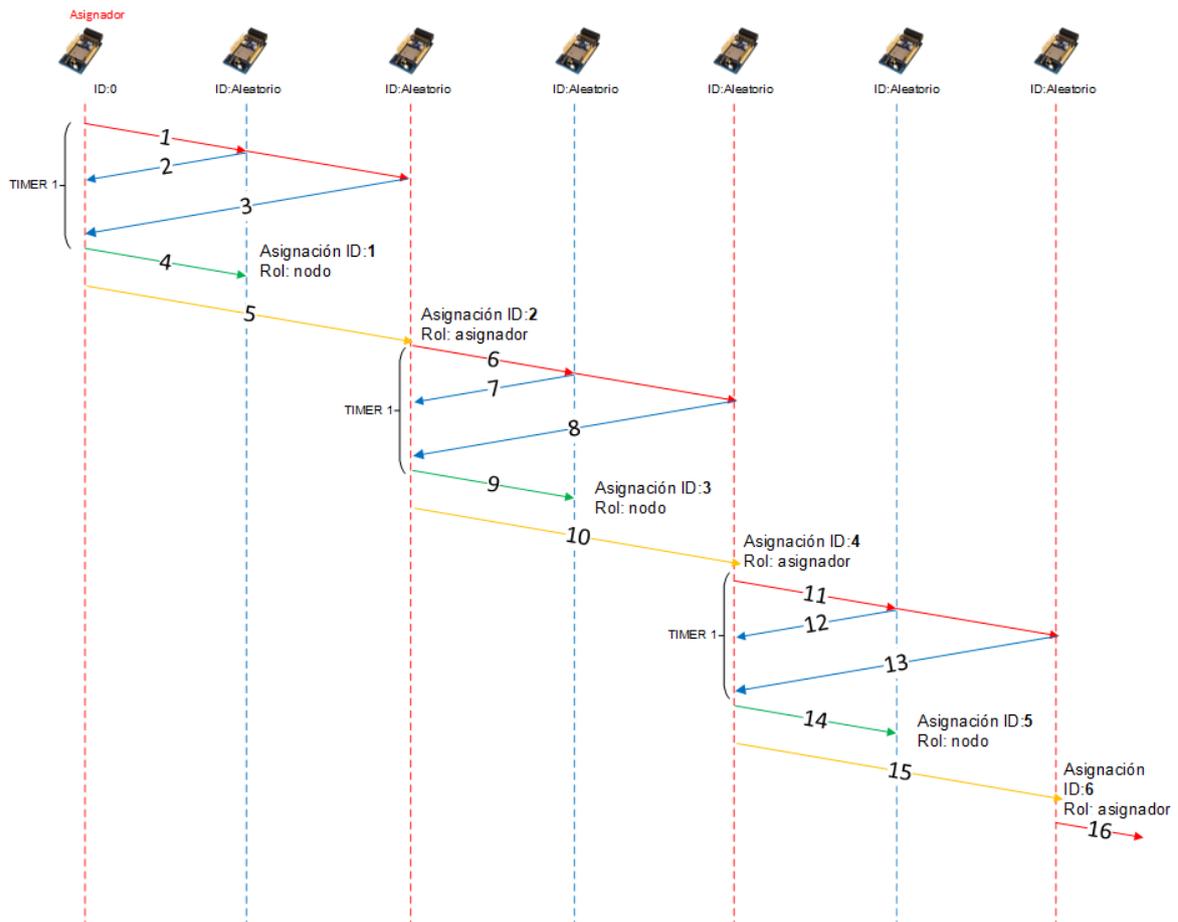


Figura 3.6. Tramas en escenario de asignación normal.

El tiempo de ejecución total del proceso de direccionamiento en un caso sin errores hasta que el último nodo borde de frontera repita el proceso de asignación se presenta en la Tabla 3.2.

Tabla 3.2. Tiempo de ejecución del algoritmo en un escenario de asignación sin errores

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
1	0	0
2	2439	2439
3	7036	4597
4	34567	27531
5	36638	2071
6	40156	3518
7	42899	2743

8	45913	3014
9	73422	27509
10	75173	1751
11	77763	2590
12	78924	1161
13	82561	3637
14	112289	29728
15	115319	3030
16	116948	1629
Tiempo total de ejecución (us)		116948

3.3.2. PRUEBA 2

Enlaces fallidos en la recepción de peticiones de asignación.

El objetivo de la prueba es comprobar el funcionamiento del algoritmo en caso de que un nodo asignador no reciba la respuesta al *broadcast* de asignación. En esta prueba no fue necesario simular mediante código un enlace fallido, debido a que es un escenario muy común. Para simular la no respuesta de un nodo vecino, se debe presionar una vez el botón SW2 del transceptor lo cual impedirá al nodo emitir la primera trama de respuesta.

Las tramas generadas en la prueba 2 por cada nodo en la red inalámbrica pueden ser observadas en la Figura 3.7.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-68	OK
RX 2	+2747 =2747	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x78D9	R	-70	OK
RX 3	+4298 =7045	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x00E0	R	-64	OK
RX 4	+27452 =34497	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x78D9	0x0000	H 1	-70	OK
RX 5	+2065 =36562	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x00E0	0x0000	Y 2	-69	OK
RX 6	+2257 =38819	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 7	+2757 =41576	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x05DF	R	-78	OK
RX 8	+30201 =71777	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 9	+2105 =73882	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x363E	R	-71	OK
RX 10	+32779 =106661	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 11	+1479 =108140	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0002	0x05DF	R	-77	OK
RX 12	+1590 =109730	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x363E	R	-72	OK
RX 13	+36882 =146612	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x05DF	0x0002	H 1	-63	OK
RX 14	+1747 =148359	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x363E	0x0002	Y 2	-63	OK
RX 15	+3220 =151579	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0004	A	-72	OK
RX 16	+2741 =154320	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0004	0x6684	R	-74	OK
RX 17	+31755 =186075	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0004	A	-72	OK
RX 18	+2412 =188487	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0004	0x6684	R	-74	OK
RX 19	+29987 =218474	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0004	A	-72	OK
RX 20	+1462 =219936	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0004	0x2ED6	R	-79	OK
RX 21	+1084 =221020	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0004	0x6684	R	-75	OK
RX 22	+32093 =253113	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x2ED6	0x0004	H 1	-72	OK
RX 23	+2062 =255175	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xCAFE	0x6684	0x0004	Y 2	-72	OK
RX 24	+2907 =258082	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0006	A	-75	OK
RX 25	+34204 =292286	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0006	A	-75	OK
RX 26	+34152 =326438	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0xFFFF	0x0006	A	-75	OK
RX 27	+16450085 =16776523	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0001	0x0001	T	-71	OK
RX 28	+297752 =17074275	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x0015	0x4075	U	-76	ERR
RX 29	+126833 =17201108	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x0003	0x0003	T	-76	OK

Figura 3.7. Captura de tramas prueba 2

En este caso se produce dos enlaces fallidos en el segundo y tercer salto de red. Como se puede observar en las tramas 7 y 9 de la Figura 3.7, solo se envía una trama de respuesta por parte de los nodos vecinos del nodo asignador 0002. Debido a esto el nodo asignador considera un enlace fallido y reenvía 3 veces la trama *broadcast* visto en las tramas con numeración 6, 8 y 9. Lo mismo sucede en el proceso de asignación de direcciones del nodo asignador 0004.

A pesar de que la trama 28 presenta un error en su lectura por parte del *sniffer*, se predice que el nodo 0005 obtuvo su identificador correctamente y envía su trama de prueba como lo hicieron los nodos 0001 y 0003 que se pueden observar en las tramas 27 y 29 respectivamente.

El proceso de asignación obtenido también se procede a describir gráficamente en la Figura 3.8.

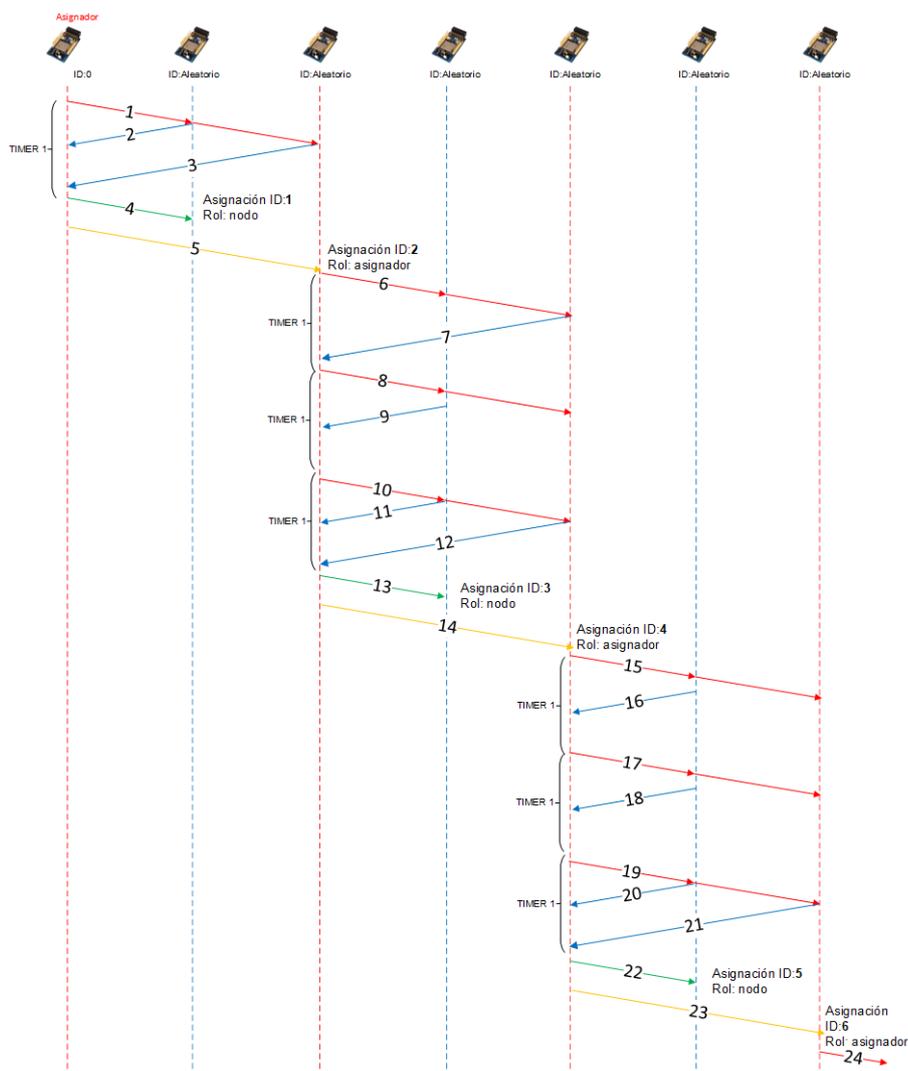


Figura 3.8. Tramas en escenario de enlaces fallidos prueba 2

El tiempo de ejecución total del proceso de direccionamiento en un caso con enlaces fallidos en la recepción de peticiones de asignación hasta que el último nodo borde de frontera repita el proceso de asignación se presenta en la Tabla 3.3.

Tabla 3.3. Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 2

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
1	0	0
2	2747	2747
3	7045	4298
4	34497	27452
5	36562	2065
6	388119	2257
7	41576	2757
8	71777	30201
9	73882	2105
10	106661	32779
11	108140	1479
12	109730	1590
13	146612	36882
14	148359	1747
15	151579	3220
16	154320	2741
17	186075	31755
18	188487	2412
19	218474	29987
20	219936	1462
21	221020	1084
22	253113	32093
23	255175	2062
24	258082	2907
Tiempo total de ejecución (us)		258082

3.3.3. PRUEBA 3

Enlace fallido en caso de que un nodo cercano o intermedio no reciba identificador

El objetivo de esta prueba es comprobar el correcto funcionamiento en caso de que un nodo cercano o intermedio no reciba la trama de direccionamiento por parte de su nodo asignador. Para simular este escenario es necesario presionar dos veces el botón SW2 del transceptor, lo cual se lo hizo en el nodo que obtendrá la dirección 0001.

Las tramas generadas por cada nodo en la red se pueden observar en la Figura 3.9

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-70	OK
RX 2	+2424 =2424	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x075E	R	-64	OK
RX 3	+970 =3394	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x6440	R	-67	OK
RX 4	+31078 =34472	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x6440	0x0000	N 1	-71	OK
RX 5	+2057 =36529	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x075E	0x0000	Y 2	-70	OK
RX 6	+2566 =39095	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 7	+1169 =40264	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x6440	R	-67	OK
RX 8	+2219 =42493	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x629E	R	-69	OK
RX 9	+771 =43254	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x55EF	R	-78	OK
RX 10	+28481 =71735	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0000	0x0002	E	-64	OK
RX 11	+3095 =74830	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x6440	0x0000	N 1	-70	OK
RX 12	+2710 =77540	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x0002	0x0000	Y 2	-70	OK
RX 13	+3207 =80747	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0002	A	-65	OK
RX 14	+1790 =82537	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x629E	R	-67	OK
RX 15	+31618 =114155	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0002	A	-65	OK
RX 16	+1788 =115943	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0002	0x629E	R	-68	OK
RX 17	+1284 =117227	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0002	0x55EF	R	-77	OK
RX 18	+38785 =156012	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x55EF	0x0002	N 1	-64	OK

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 19	+1108 =157120	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0x629E	0x0002	Y 2	-64	OK
RX 20	+1635 =158755	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xC9FE	0xFFFF	0x0004	A	-68	OK
RX 21	+1791 =160546	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xC9FE	0x0004	0x2FE8	R	-76	OK
RX 22	+32076 =192622	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xC9FE	0xFFFF	0x0004	A	-69	OK
RX 23	+1148 =193770	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xC9FE	0x0004	0x0854	R	-80	OK
RX 24	+1601 =195371	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xC9FE	0x0004	0x2FE8	R	-78	OK
RX 25	+30761 =226132	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xC9FE	0x0854	0x0004	N 1	-69	OK
RX 26	+2421 =228553	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0x2FE8	0x0004	Y 2	-69	OK
RX 27	+2600 =231153	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xC9FE	0xFFFF	0x0006	A	-78	OK
RX 28	+34607 =265760	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xC9FE	0xFFFF	0x0006	A	-79	OK
RX 29	+34839 =300599	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xC9FE	0xFFFF	0x0006	A	-78	OK
RX 30	+16516609 =16817208	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xC9FE	0x0001	0x0001	T	-60	OK
RX 31	+31366 =16848574	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xC9FE	0x0005	0x0005	T	-81	OK
RX 32	+359968 =17208542	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xC9FE	0x0003	0x0003	T	-80	OK

Figura 3.9. Captura de tramas prueba 3

El nodo 0002 al recibir tres tramas de respuesta de asignación (tramas 7, 8, 9) detecta que existen más nodos dentro de su área de cobertura debido a que un nodo vecino anterior no recibió su identificador, por lo que procede a enviar una trama unicast (trama 10) con la etiqueta "E" en el *payload* al nodo asignador anterior, el cual reenvía las tramas con información de direccionamiento (tramas 11 y 12) para proceder a identificar correctamente al nodo 0001 y para que el nodo 0002 continúe con el proceso de asignación.

Además, se puede observar de nuevo el escenario de enlaces fallidos obtenido en la prueba 2, dado el reenvío de las tramas *broadcast* (tramas 15 y 22) por parte de los nodos asignadores 0002 y 0004.

La correcta asignación de los identificadores 0001, 0003 y 0005 de los nodos vecinos intermedios puede ser vista en la dirección de origen de las tramas de prueba 30, 31 y 32.

Lo descrito anteriormente se presenta en la Figura 3.10

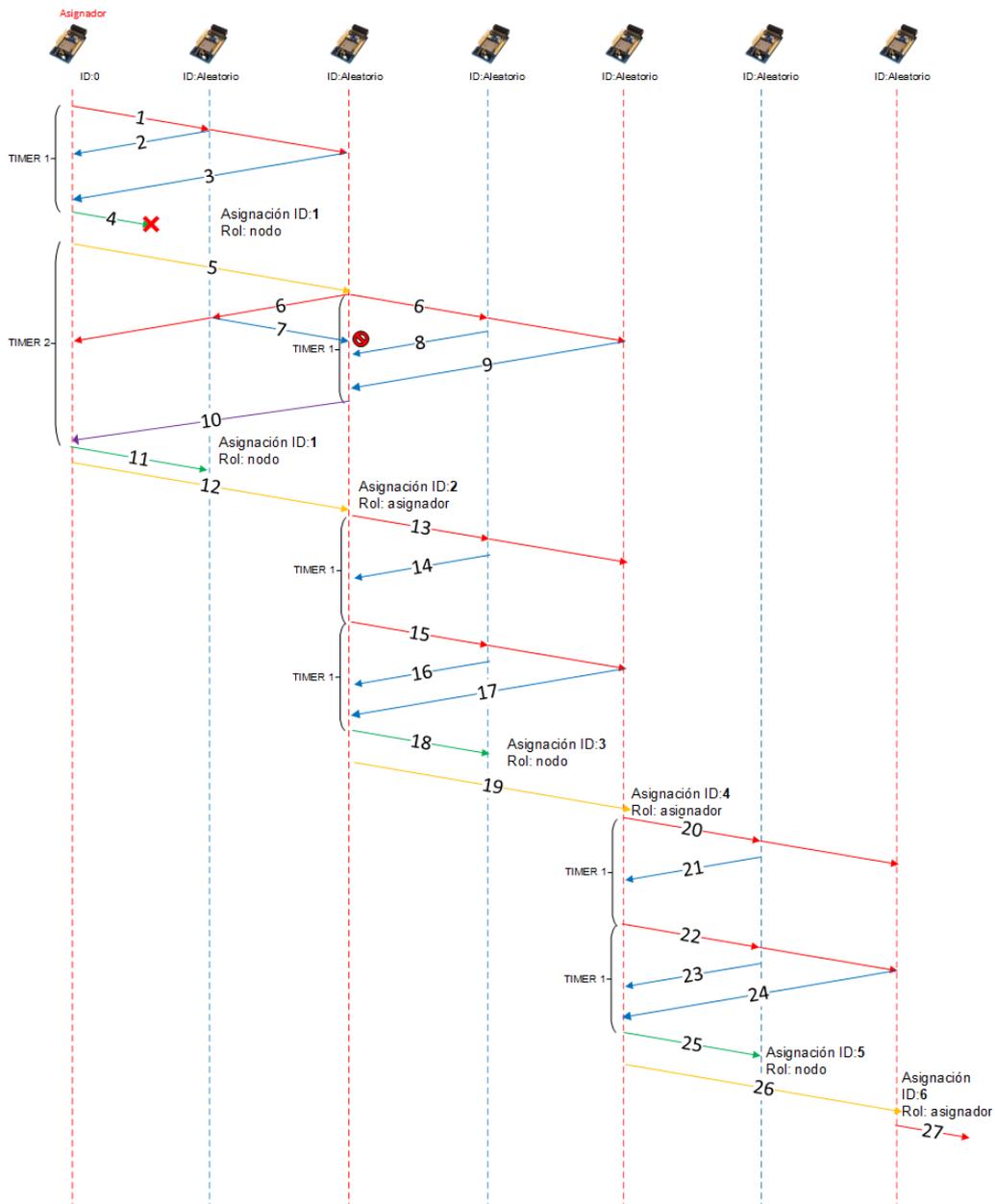


Figura 3.10 Tramas en escenario de enlaces fallidos prueba 3

El tiempo de ejecución total del proceso de direccionamiento en un caso con enlaces fallidos cuando un nodo cercano o intermedio no recibe su identificador hasta que el último nodo de frontera repite el proceso de asignación se presenta en la Tabla 3.4.

Tabla 3.4 Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 3

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
-------	-----------------------------------	--

1	0	0
2	2424	2424
3	3394	970
4	34472	31078
5	36529	2057
6	39095	2566
7	40264	1169
8	42483	2219
9	43254	771
10	71735	28481
11	74830	3095
12	77540	2710
13	80747	3207
14	82537	1790
15	114155	31618
16	115943	1788
17	117227	1284
18	156012	38785
19	157120	1108
20	158755	1635
21	160546	1791
22	192622	32076
23	193770	1148
24	195371	1601
25	226132	30761
26	228553	2421
27	231153	2600
Tiempo total de ejecución (us)		231153

3.3.4. PRUEBA 4

Enlace fallido en caso de que un nodo de borde no reciba identificador

El objetivo de la prueba es validar el funcionamiento en caso de que un enlace fallido se produzca en la trama de asignación enviada por parte de un nodo asignador a un nodo que será configurado como siguiente nodo asignador en la red. Para simular este escenario es necesario presionar dos veces el botón SW2 del transceptor, esto se lo hizo en el nodo de borde que obtendrá la dirección 0002.

La Figura 3.11 contiene las tramas capturadas por el *sniffer* y las cuales se presentan a continuación.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-69	OK
RX 2	+1789 =1789	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x535F	R	-68	OK
RX 3	+1268 =3057	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x25DB	R	-64	OK
RX 4	+31427 =34484	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x535F	0x0000	N 1	-69	OK
RX 5	+3334 =37818	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x25DB	0x0000	Y 2	-69	OK
RX 6	+68558 =106376	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x25DB	0x0000	Y 2	-70	OK
RX 7	+4628 =111004	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0002	A	-65	OK
RX 8	+3056 =114060	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x72DD	R	-70	OK
RX 9	+31171 =145231	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0002	A	-65	OK
RX 10	+1155 =146386	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x6850	R	-81	OK
RX 11	+2238 =148624	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x72DD	R	-69	OK
RX 12	+29938 =178562	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x6850	0x0002	N 1	-65	OK
RX 13	+1437 =179999	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x72DD	0x0002	Y 2	-64	OK

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+2592	12	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x0004	A	-70	OK
	=182591		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+1162	12	Type Sec Pnd Ack.req PAN_compr	0x00	0xCAFE	0x0004	0x6F30	R	-79	OK
	=183753		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+4596	12	Type Sec Pnd Ack.req PAN_compr	0x00	0xCAFE	0x0004	0x0636	R	-77	OK
	=188349		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+27179	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x6F30	0x0004	N	-70	OK
	=215528		DATA 0 0 0 1					1		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+2065	13	Type Sec Pnd Ack.req PAN_compr	0x04	0xCAFE	0x0636	0x0004	Y	-70	OK
	=217593		DATA 0 0 0 1					2		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+2265	12	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0xFFFF	0x0006	A	-76	OK
	=219858		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+2065	12	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x0006	A	-76	OK
	=252464		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+33579	12	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0xFFFF	0x0006	A	-75	OK
	=286043		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+16493329	12	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0x0001	0x0001	T	-71	OK
	=16779372		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+257540	12	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0x0005	0x0005	T	-78	OK
	=17036912		DATA 0 0 0 1							
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX	+196237	12	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0x0003	0x0003	T	-78	OK
	=17233149		DATA 0 0 0 1							

Figura 3.11. Captura de tramas prueba 4

En la Figura 3.12 que explica gráficamente el proceso de asignación, se puede observar cómo el próximo nodo asignador 0002 no recibe la trama de asignación (trama 5) ocasionando que no pueda ser configurado como siguiente nodo asignador, el nodo 0000 al no detectar el *broadcast* de asignación del siguiente nodo asignador vuelve a emitir la trama de asignación (trama 6).

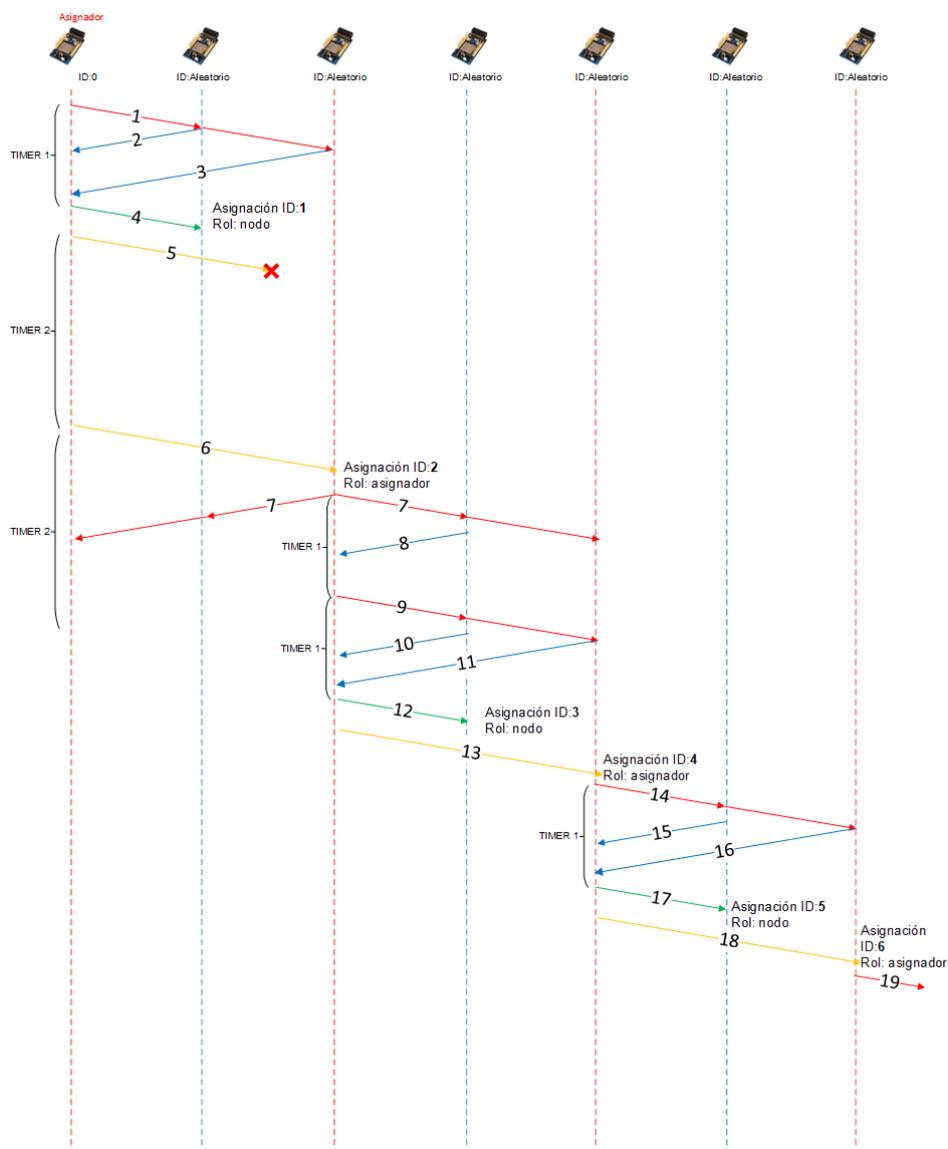


Figura 3.12. Tramas en escenario de enlaces fallidos prueba 4

El tiempo de ejecución total del proceso de direccionamiento en un caso con enlaces fallidos cuando un nodo de borde no recibe su identificador hasta que el último nodo borde de frontera repite el proceso de asignación se presenta en la Tabla 3.5.

Tabla 3.5. Tiempo de ejecución del algoritmo en un escenario de enlaces fallido prueba 4

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
1	0	0
2	1789	1789
3	3057	1268

4	34484	31427
5	37818	3334
6	106376	68558
7	111004	4628
8	114060	3056
9	145231	31171
10	146386	1155
11	148624	2238
12	178562	29938
13	179999	1437
14	182591	2592
15	183753	1162
16	188349	4596
17	215528	27179
18	217593	2065
19	219858	2265
Tiempo total de ejecución (us)		219858

3.3.5. PRUEBA 5

Nodo cercano o intermedio caído.

El objetivo de la prueba es validar el funcionamiento y correcta asignación de direcciones en caso de que en la red con topología lineal se presente un nodo intermedio fallido. Para la prueba se procederá a desconectar las baterías de un nodo intermedio, el cual al no estar encendido, no responderá al *broadcast* de detección enviado por el nodo asignador y no obtendrá su dirección.

Cabe recalcar lo mencionado en el apartado 3.2. de pruebas de medición de potencia, para el caso de nodos fallidos es necesario encontrar un umbral adecuado para discernir correctamente la dirección que se reservará al nodo fallido. Para la prueba se configuró un umbral de potencia de -50 dBm.

La Figura 3.13 contiene las tramas capturadas por el *sniffer* en la prueba 5, las cuales se presentan a continuación.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-73	OK
RX 2	+2749 =2749	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x188A	R	-63	OK
RX 3	+30120 =32869	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0000	A	-71	OK
RX 4	+3395 =36264	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x188A	R	-64	OK
RX 5	+21680 =57944	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0000	A	-70	OK
RX 6	+3059 =61003	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0000	0x188A	R	-64	OK
RX 7	+30301 =91304	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x188A	0x0000	Y	-71	OK
RX 8	+1931 =93235	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 9	+3080 =96315	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x337E	R	-76	OK
RX 10	+31467 =127782	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0002	A	-64	OK
RX 11	+1153 =128935	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0002	0x337E	R	-77	OK
RX 12	+1920 =130855	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0002	0x3E4D	R	-73	OK
RX 13	+38221 =169076	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x337E	0x0002	N 1	-63	OK
RX 14	+2390 =171466	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xCAFE	0x3E4D	0x0002	Y 2	-64	OK
RX 15	+1619 =173085	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0004	A	-72	OK
RX 16	+1783 =174868	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0004	0x030F	R	-78	OK
RX 17	+951 =175819	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0004	0x36B9	R	-82	OK
RX 18	+30208 =206027	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x36B9	0x0004	N 1	-73	OK
RX 19	+1112 =207139	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x030F	0x0004	Y 2	-73	OK
RX 20	+1321 =208460	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0006	A	-78	OK
RX 21	+34905 =243365	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0006	A	-77	OK
RX 22	+33553 =276918	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0006	A	-77	OK
RX 23	+16551331 =16828249	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0005	0x0005	T	-85	OK
RX 24	+395969 =17224218	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0003	0x0003	T	-77	OK
RX 25	+7732646 =24956864	29	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0000	MAC payload ID no asi gnada : 1	-70	OK

Figura 3.13. Captura de tramas prueba 5

Se puede observar en la Figura 3.13 que el nodo 0000 reenvía tres veces la trama *broadcast* de detección (tramas 1, 3, 5) y solo recibe respuesta de un solo nodo, por lo que procede a enviar solo un identificador (trama 7). También se observa la presencia de un enlace fallido en el segundo salto de red. La correcta asignación de los identificadores 0003 y 0005 de los nodos vecinos intermedios puede ser vista en la dirección de origen de las tramas de prueba 23 y 24. Además, en la trama 25 el nodo envía en su *payload* el identificador que no pudo ser asignado.

Sin contar con las tramas de prueba, lo explicado anteriormente se resume de manera gráfica en la Figura 3.14.

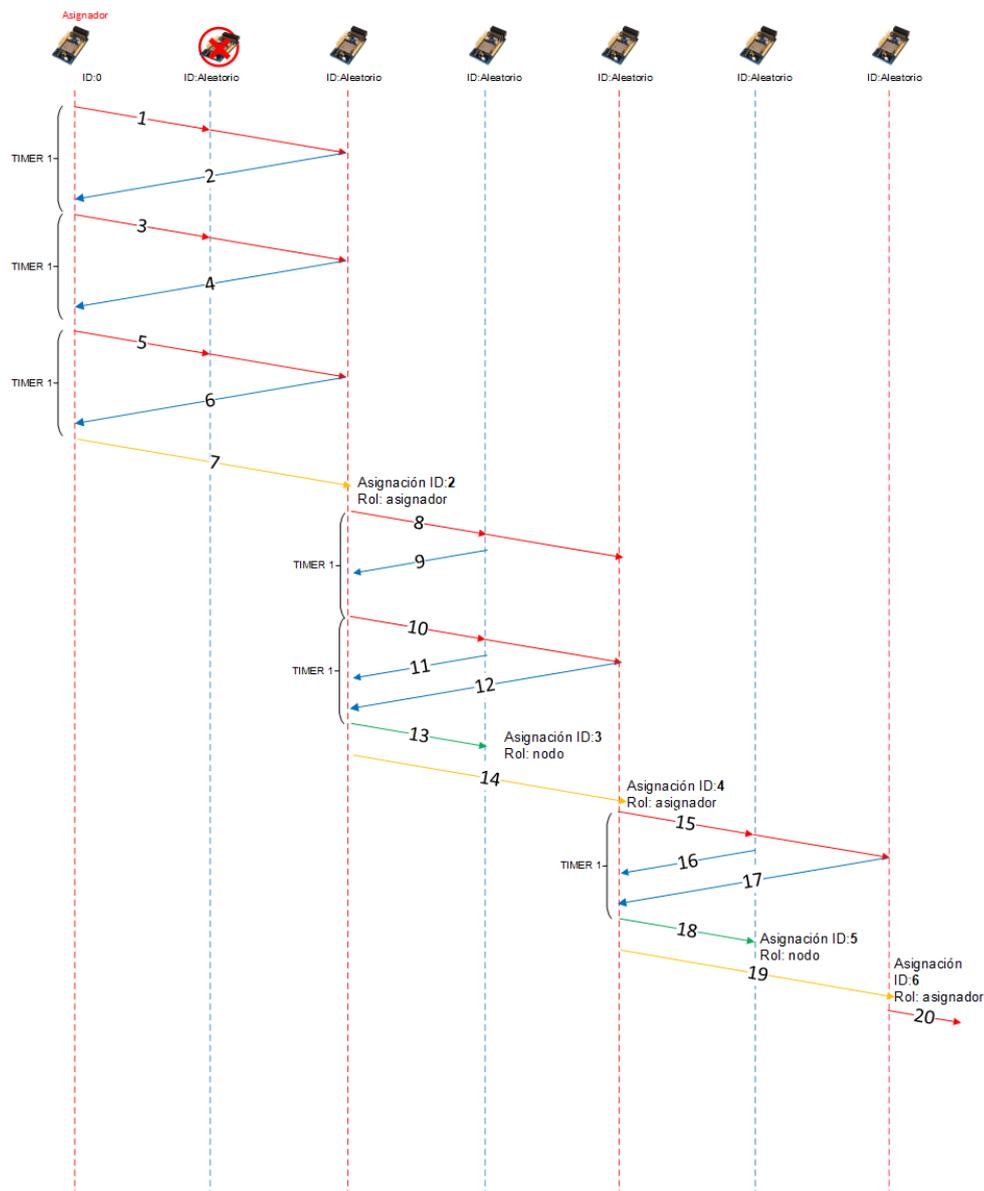


Figura 3.14. Tramas en escenario de enlaces fallidos prueba 4

El tiempo de ejecución total del proceso de direccionamiento en caso de que un nodo intermedio se encuentre averiado hasta que el último nodo de frontera repita el proceso de asignación se presenta en la Tabla 3.6

Tabla 3.6. Tiempo de ejecución del algoritmo en un escenario de nodo fallido prueba 5

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
1	0	0
2	2749	2749
3	32869	30120
4	36264	3395
5	57944	21680
6	61003	3059
7	91304	30301
8	93235	1931
9	96315	3080
10	127782	31467
11	128935	1153
12	130855	1920
13	169076	38221
14	171476	2390
15	173085	1619
16	174868	1783
17	175819	951
18	206027	30208
19	207139	1112
20	208460	1321
Tiempo total de ejecución (us)		208460

3.3.6. PRUEBA 6

Nodo de borde o frontera caído.

Para la prueba 6 se procede a desconectar dos nodos de borde, los cuales no se convertirán en nuevos nodos asignadores, el nodo caído 0002 no responderá al *broadcast* de asignación ocasionando que el nodo intermedio 0001 sea nombrado nuevo nodo asignador, por lo que este nodo asignará las direcciones a partir del nodo 0003 continuando con el proceso hasta la asignación completa de identificadores en todos los nodos de red. El nodo 0005 al no detectar respuesta del último nodo caído 0006 finalizará el proceso de asignación de direcciones.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 1	+0 =0	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0xFFFF	0x0000	A	-69	OK
RX 2	+2414 =2414	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0000	0x7BF8	R	-69	OK
RX 3	+32898 =35312	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0000	A	-70	OK
RX 4	+2423 =37735	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x7BF8	R	-69	OK
RX 5	+30880 =68615	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0000	A	-70	OK
RX 6	+2740 =71355	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0000	0x7BF8	R	-69	OK
RX 7	+31138 =102493	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x7BF8	0x0000	Y 1	-69	OK
RX 8	+3175 =105668	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0001	A	-69	OK
RX 9	+2417 =108085	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0001	0x5326	R	-91	OK
RX 10	+29845 =137930	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0xFFFF	0x0001	A	-69	OK
RX 11	+1155 =139085	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0001	0x5326	R	-88	OK
RX 12	+33576 =172661	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0xFFFF	0x0001	A	-69	OK
RX 13	+1171 =173832	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0001	0x5326	R	-90	OK

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	RSSI (dBm)	FCS
RX 14	+38225 =212057	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xCAFE	0x5326	0x0001	Y 2	-69	OK
RX 15	+3206 =215263	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0003	A	-89	OK
RX 16	+2750 =218013	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0003	0x546C	R	-86	OK
RX 17	+5250 =223263	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x0003	0x59F8	R	-73	OK
RX 18	+25052 =248315	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x59F8	0x0003	N 1	-86	OK
RX 19	+1126 =249441	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x546C	0x0003	Y 2	-87	OK
RX 20	+2256 =251697	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0005	A	-85	OK
RX 21	+45084 =296781	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x0005	A	-85	OK
RX 22	+36045 =332826	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0xFFFF	0x0005	A	-85	OK
RX 23	+8328611 =8661437	19	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xCAFE	0xEFFF	0x0003	-44# -55#	-94	OK
RX 24	+8212303 =16873740	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0004	0x0004	I	-76	OK
RX 25	+8233202 =25106942	29	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x07	0xCAFE	0xFFFF	0x0001	ID no asi gnada : 2	-69	OK

Figura 3.15. Captura de tramas prueba 6

En la captura de tramas de la Figura 3.15 se puede observar que el nodo 0000 reenvia tres veces la trama *broadcast* de detección (tramas 1,3,5), al no recibir respuesta por parte del nodo caído de borde procede a enviar solo al nodo cercano su identificador (trama 7). El siguiente nodo asignador 0001 también detecta la presencia del nodo caído después de retransmitir tres veces la trama *broadcast* de detección (tramas 8,10,12) por lo que procede a identificar solo a su nodo de borde 0003 (trama 14). Después, el nodo de borde 0003 procede a repetir el proceso de asignación enviando las tramas de asignación (tramas 18,19) a los nodos que obtendrán la identificación 0004 y 0005. El nodo 0005 al no recibir respuesta del siguiente nodo caído finaliza el proceso.

En la trama de prueba con secuencia 23 de la Figura 3.15 se puede observar los niveles de señal que recibió el nodo 0003 por parte de sus nodos vecinos. Este tipo de tramas se las puede obtener quitando el comentario a las variables `id_proceso` y `procesamiento` que permiten enviar en el *payload* de la trama de prueba los niveles de señal recibida. Ver Código 3.1.

```
//Si ya finalizó el proceso de asignación.
if(id_proceso==ASIGNACION_NORMAL && retransmision == 0){
    //id_proceso=ENVIO_RESUMEN_POTENCIA_ASIGNACION;    //Para medición de nivel de potencia
    //procesamiento=true;                               //Activar el procesamiento
```

Código 3.1. Código de medición de potencia recibida. Código comentado

En la trama de prueba 24 se puede observar que el nodo 0004 obtuvo correctamente su identificador. Por último, en el *payload* de la trama de prueba 25 se puede observar el identificador que no fue asignado por el nodo 0001.

Sin contar con las tramas de prueba, el proceso de asignación en caso de nodos fallidos explicado anteriormente se resume de manera gráfica en la Figura 3.16.

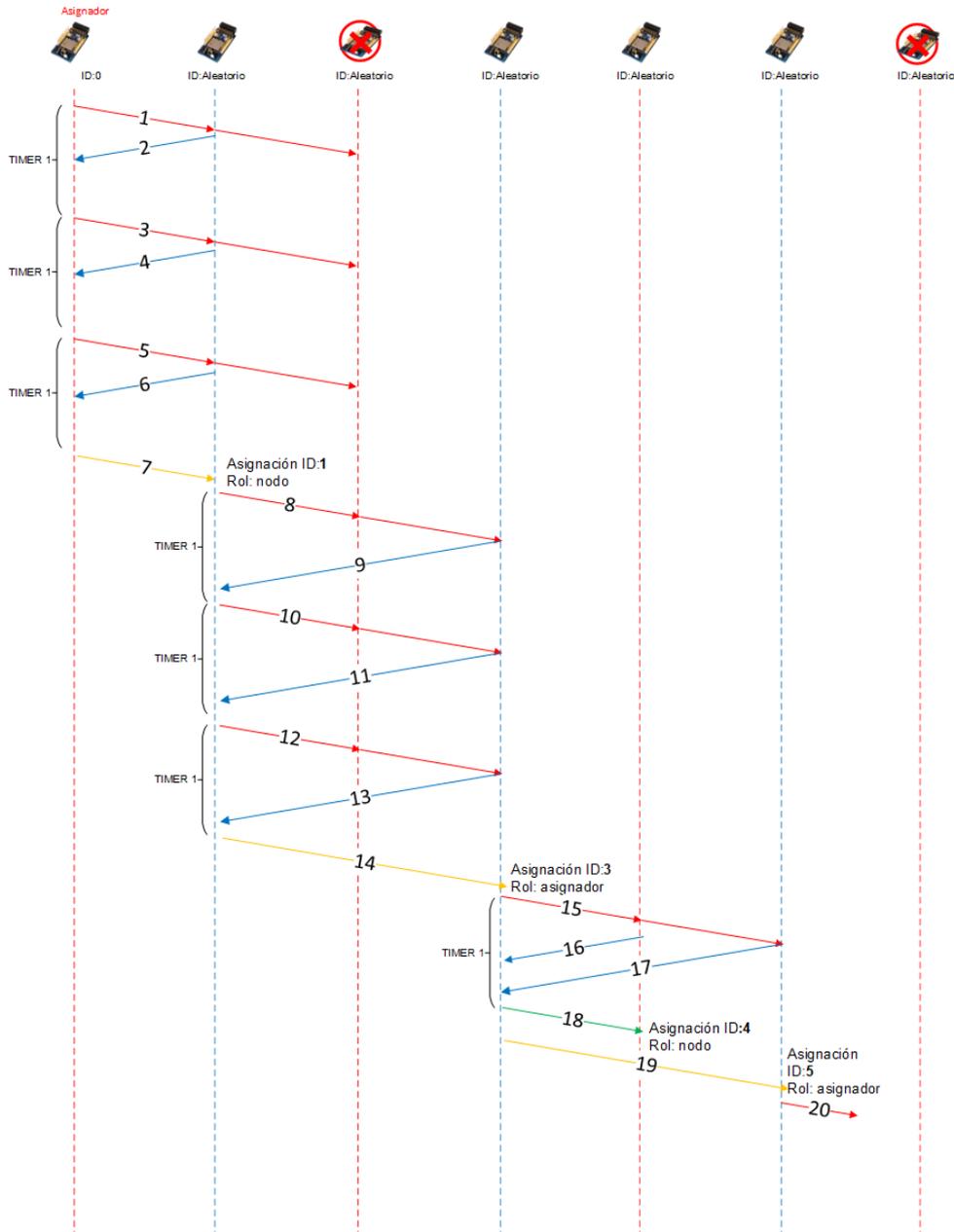


Figura 3.16. Tramas en escenario de enlaces fallidos prueba 4

El tiempo de ejecución total del proceso de direccionamiento en caso de que un nodo de borde se encuentre averiado hasta que el último nodo de frontera repita el proceso de asignación se presenta en la Tabla 3.7.

Tabla 3.7. Tiempo de ejecución del algoritmo en un escenario de nodo fallido prueba 5

Trama	Tiempo de captura (microsegundos)	Intervalo de tiempo entre tramas (microsegundos)
1	0	0
2	2414	2414
3	35312	32898
4	37735	2423
5	68615	30880
6	71355	2740
7	102493	31138
8	105668	3175
9	108085	2417
10	137930	29845
11	139085	1155
12	172661	33576
13	173832	1171
14	212057	38225
15	215263	3206
16	218013	2750
17	223263	5250
18	248315	25052
19	249441	1126
20	251697	2256
Tiempo total de ejecución (us)		251697

3.4. COMPARACIÓN DE TIEMPOS

3.4.1 COMPARACIÓN CON TIEMPO TEÓRICO.

Usando la Ecuación 2.1 y la Ecuación 2.3 explicadas en la sección 2.3, para un total de siete nodos presentes en la red con topología lineal multisalto, con dos nodos vecinos dentro de la zona de cobertura de un nodo asignador; similar al escenario de prueba de asignación sin errores usado en las pruebas experimentales, obtenemos un tiempo teórico total de asignación de **115086 microsegundos** con un Timer 1 de 31250 microsegundos. Ver Ecuación 3.1

$$T_{ASIGTOTAL} = \left[\frac{x}{n} \right] (T_1 + n T_n) = 3 (31250 + 2(3556)) = 115086 \text{ us}$$

Ecuación 3.1. Cálculo de tiempo teórico.

Cabe recalcar que el tiempo de espera T1 es igual al tiempo del Timer 1 configurado previamente para esperar las tramas de respuesta por parte de los nodos vecinos, el cual se lo obtuvo de manera experimental realizando varias pruebas.

El tiempo total de ejecución obtenido de manera experimental para un escenario sin errores en la asignación fue de **116948 microsegundos**, el cual se lo puede observar en la Tabla 3.2.

3.4.2 COMPARACIÓN DE TIEMPOS CON OTROS PROTOCOLOS.

El objetivo de la comparación es evaluar la propuesta de tiempo de asignación respecto a otras tecnologías como 6LoWPAN y ZigBee obteniendo datos de pruebas experimentales.

Dado lo explicado en la sección 1.3.3 sobre ZigBee en topologías lineales, se justifica que el protocolo ZigBee no funciona en condiciones similares al prototipo de red propuesto, por lo que no puede ser comparado adecuadamente con el mismo.

3.4.2.1 Comparación con protocolo 6LoWPAN.

Para la comparación del algoritmo desarrollado con el protocolo 6LoWPAN se usaron datos de tesis de titulación elaboradas, que cumplieran con condiciones similares; como distancias cortas de separación y el número de nodos usados.

En el trabajo de titulación "Optimización de flujo de datos en un prototipo de red 6LoWPAN con topología lineal" [38]. Se usó un prototipo de red 6LoWPAN similar al usado en el presente trabajo de titulación. Ver Figura 3.17.

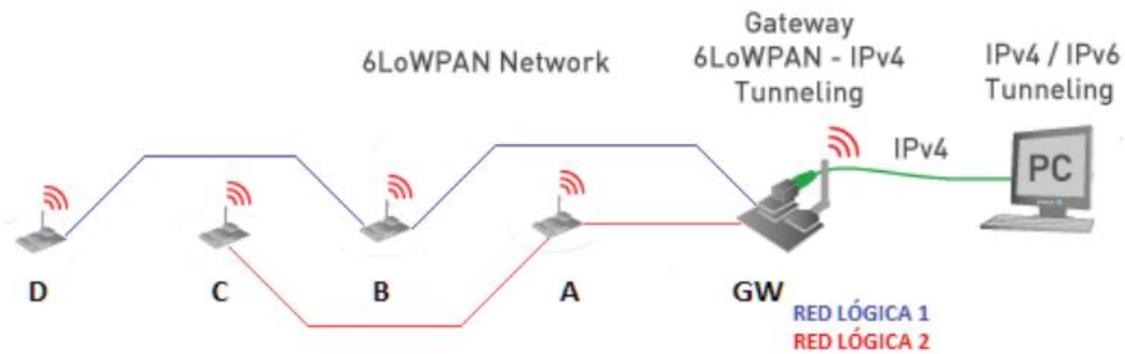


Figura 3.17. Prototipo 1 de tesis de red 6LoWPAN [38]

La Tabla 3.8. muestra los tiempos comparativos obtenidos al realizar pruebas en un red 6LoWPAN con topología lineal multisalto. Los datos son extraídos del trabajo de titulación mencionado, en el cual se explica cómo se obtuvieron los tiempos de recuperación en caso de un nodo fallido en la red, similar al escenario de Prueba 5 explicado en la sección 3.3.

Tabla 3.8. Tiempos obtenidos de una red 6LoWPAN [38]

Tiempo de recuperación	ALGORITMO [segundos]	SIN ALGORITMO [segundos]
1	63	2
2	39	27
3	44	7
Tiempo promedio	48,67	12

Adicionalmente, en el trabajo de titulación “Caracterización de retardos de 6LoWPAN en redes lineales a gran escala a partir de datos experimentales obtenidos en una red escala reducida” [39], se realizaron diferentes pruebas experimentales para obtener los tiempos de establecimiento de la red.

Los datos extraídos y encontrados en dicho trabajo de titulación muestran un tiempo promedio de establecimiento de la red de **23.57 segundos** usando nodos 6LoWPAN del fabricante *MEMSIC* y un tiempo promedio de **49.362219 segundos** usando nodos 6LoWPAN del fabricante *Libelium*.

Los parámetros similares de comparación con los cuales se acopló el prototipo al trabajo de titulación mencionado anteriormente son: distancias cortas entre los nodos (2.5 metros) y un total de 6 nodos en la red.

Para el algoritmo desarrollado de asignación de direcciones para una topología lineal, se presenta en la Tabla 3.9 un resumen de tiempos de asignación para todos los escenarios de prueba, usando un total de 7 nodos. Los datos son extraídos de las tablas de tiempos de ejecución del algoritmo encontrados en la sección 3.3.

Tabla 3.9 Resumen de tiempos de asignación del algoritmo desarrollado.

Prueba	Escenario	Tiempo de asignación (segundos)
1	Asignación Normal	0.116948
2	Enlaces Fallidos 1	0.258082
3	Enlaces Fallidos 2	0.231153
4	Enlaces Fallidos 3	0.219858
5	Nodos Caídos 1	0.20846
6	Nodos Caídos 2	0.251697

3.5. ANÁLISIS DE LOS RESULTADOS

Después de haber culminado las pruebas se pudo observar los siguientes resultados:

El algoritmo desarrollado identifica y asigna direcciones a todos los nodos presentes en la topología de red lineal multisalto.

El algoritmo mantiene el direccionamiento de los nodos inalámbricos en caso de presentarse errores en la comunicación entre los nodos, como enlaces fallidos o nodos caídos.

El algoritmo presenta limitación en el número de nodos vecinos que pueden estar presentes en el área de cobertura de un nodo asignador, debido al uso de la señal de potencia recibida para clasificar las direcciones por parte de los nodos asignadores, resultado obtenido en las pruebas de medición de potencia.

El *sniffer* utilizado funcionó adecuadamente y logró captar todas las tramas generadas en el proceso de direccionamiento, sin embargo, puede presentar lectura de tramas erróneas como la observada en la prueba 2.

En las pruebas con nodos fallidos se evidencia la necesidad de encontrar un umbral de potencia adecuado, caso contrario el nodo asignador podría confundir a un nodo

intermedio con uno de borde en el envío de identificadores. Esta limitante no ocasiona que los nodos no reciban identificadores.

El tiempo experimental de ejecución del algoritmo de asignación de direcciones, en un escenario sin errores en la comunicación, es muy similar al tiempo calculado de manera teórica.

El tiempo de asignación del algoritmo de direccionamiento desarrollado, es mucho menor al tiempo que le toma al protocolo 6LoWPAN establecer la red asignando direcciones. Resultado que evidencia claramente la ventaja del algoritmo desarrollado frente a otro protocolo.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

El desarrollo del algoritmo de asignación de direcciones globales en una topología lineal que utiliza el nivel de enlace del estándar IEEE 802.15.4 permite reducir los tiempos de retardo en el proceso de asignación de direcciones debido a que se aprovecha la simplicidad de la topología de red reduciendo los procesos innecesarios en el control de la red.

Es posible adaptar, desarrollar e implementar código a nivel de capa de enlace usando un transceptor IEEE 802.15.4 y lenguaje de programación C/C++, lo que demuestra la flexibilidad de crear cualquier aplicación que permita optimizar cualquier proceso como el de asignación de direcciones.

La ventaja de la propuesta se evidencia claramente en los tiempos de asignación producidos por el algoritmo desarrollado los cuales son mucho menores a los comparados con otras tecnologías como 6LoWPAN, lo que abre la posibilidad de realizar aplicaciones de monitoreo de bajo costo aplicadas a redes lineales multsalto usando solo el estándar IEEE 802.15.4.

Las pruebas realizadas demostraron que el algoritmo implementado también previene situaciones de fallo en la comunicación de la red, tales como enlaces fallidos y nodos caídos obteniendo también de esta manera un grado de confiabilidad en el establecimiento de la red.

A pesar de que el algoritmo está diseñado para asignar direcciones a n nodos vecinos dentro del área de cobertura de un nodo asignador, se encontró la limitante de usar solo

dos nodos vecinos por el requerimiento de reservar adecuadamente la dirección en caso de que un nodo se encuentre fallido; dado que el algoritmo usa umbrales de potencia para clasificar las señales recibidas y encontrar la posición del nodo caído. Sin contar con la presencia de nodos caídos el algoritmo no presentaría esta limitante y podría asignar a una mayor cantidad de nodos en el área de cobertura de un nodo asignador.

La proyección teórica ideal del retardo para una red lineal a gran escala con cientos de nodos también permite determinar que el algoritmo solo necesitaría unos pocos segundos para asignar identificadores a una gran cantidad de nodos en una topología lineal multisalto, pero en la práctica también influirían otros valores como el tiempo de propagación dependiente de la distancia entre nodos y el tiempo de acceso al canal.

4.2. RECOMENDACIONES

Es recomendable tener línea de vista entre los nodos sensores, ya que si los nodos ATZB se encuentran en el piso su alcance de transmisión disminuye. Además, es necesario ubicar adecuadamente su antena incorporada para tener una mejor comunicación entre los nodos.

Se recomienda definir el rango de cobertura entre los nodos al momento de crear la topología lineal de prueba, dado que los nodos requieren estar presentes dentro del rango de cobertura del nodo asignador para recibir su identificador.

Se recomienda calibrar el umbral de potencia de acuerdo al escenario en el que se va a probar el prototipo de red, de lo contrario se podría perder la secuencialidad de identificación en los nodos inalámbricos.

Se puede recomendar que si no se desea mantener el identificador de un nodo caído, se elimine este tipo de caso para no desperdiciar direcciones en la topología lineal, lo cual sería una posible mejora del algoritmo.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] I. F. Akyildiz y M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.
- [2] P. Rawat, K. D. Singh, H. Chaouchi, y J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies”, *J Supercomput*, vol. 68, núm. 1, pp. 1–48, abr. 2014.
- [3] F. Vásconez, “Tecnología Informática: Redes de sensores inalámbricos”, *Tecnología Informática*, el 24 de abril de 2013. http://blogfabianv.blogspot.com/2013/04/redes-de-sensores-inalambricos-una-gran_24.html (consultado el 13 de marzo de 2022).
- [4] R. Fernandez *et al.*, *Redes Inalámbricas de sensores: teoría y aplicación práctica*. 2009.
- [5] M. O. E. Alberto, “PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR FACULTAD DE INGENIERÍA MAESTRÍA EN REDES DE COMUNICACIONES”, p. 129.
- [6] J. Zheng y A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. John Wiley & Sons, 2009.
- [7] Y. Xiao y Y. Pan, *Emerging Wireless LANs, Wireless PANs, and Wireless MANs: IEEE 802.11, IEEE 802.15, 802.16 Wireless Standard Family*. Wiley, 2009.
- [8] LAN/MAN Standards Committee of the IEEE Computer Society, Part 15.4, “Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (LR-WPAN),” IEEE Computer Society, Oct. 2002
- [9] C. Buratti, M. Martalo’, R. Verdone, y G. Ferrari, *Sensor Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity*. Springer Science & Business Media, 2011.
- [10] D. J. A. Gutierrez, “Enabling Pervasive Wireless Sensor Networks”, p. 79, 2005.
- [11] S. C. Ergen, “ZigBee/IEEE 802.15.4 Summary”, p. 37.
- [12] J. A. Gutiérrez, E. H. Callaway, y R. L. Barrett, *Low-rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4*. IEEE Standards Association, 2004.
- [13] “IEEE 802.15.4 Standard: Primer Electronics Notes”. <https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/basics-tutorial-primer.php> (consultado el 18 de junio de 2022).

- [14] N. S. Bhat, "Design and Implementation of IEEE 802.15.4 Mac Protocol on FPGA", p. 5, 2011.
- [15] F. D. C. Reyes, "IMPLEMENTACIÓN DEL ALGORITMO DE PROTOCOLO DE DIRECCIONAMIENTO PARA REDES DE SENSORES INALÁMBRICOS CON EL ESTANDAR IEEE 802.15.4.", Quito: Escuela Politécnica Nacional, 2018. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/19061>. [Accedido: 13-mar-2022]
- [16] J. T. Adams, "An Introduction to IEEE STD 802.15.4", in IEEE Aerospace Conference, Big Sky, MT, USA, 2006, pp. 1-8.
- [17] Jin-Shyan Lee, "An Experiment on Performance Study of IEEE 802.15.4 Wireless Networks", in IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy, 2005, vol. 2, pp. 451-458.
- [18] M. C. Acosta, "ESTUDIO DEL ESTÁNDAR IEEE 802.15.4 "ZIGBEE" PARA COMUNICACIONES DE ÁREA PERSONAL DE BAJO CONSUMO DE ENERGÍA Y SU COMPARACIÓN CON EL ESTÁNDAR IEEE 802.15.1 "BLUETOOTH" ", Quito: Escuela Politécnica Nacional, 2006. [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/55/1/CD-0024.pdf>. [Accedido: 15-mar- 2022]
- [19] J. P. Dignani, "Análisis del Protocolo ZigBee", Tesis Doctoral, 2012.
- [20] P. Lea, *INTERNET OF THINGS FOR ARCHITECTS*, 1ra ed. Birmingham: Packt Publishing Ltd, 2018.
- [21] NXP Laboratories, "IEEE 802.15.4 Stack User Guide", vol. 2.6, 2016, [Accedido: 16-mar- 2022] [En línea]. Disponible en: <https://www.nxp.com/docs/en/user-guide/JN-UG-3024.pdf>
- [22] P. Dhillon y D. H. Sadawarti, «A Review Paper on Zigbee (IEEE 802.15.4) Standard», International Journal of Engineering Research, vol. 3, n. ° 4, p. 6, abr. 2014.
- [23] A. Elahi y A. Gschwender, *ZIGBEE WIRELESS SENSOR AND CONTROL NETWORK*, 2da ed. Boston: Pearson Education, 2009.
- [24] "Networking methods - ZigBee Networking Guide". Disponible en: <https://development.libelium.com/zigbee-networking-guide/networking-methods>. [Accedido: 21-mar- 2022]

- [25] N. Saholia y S. Joshi, "Review on address assignment mechanism in ZigBee wireless sensor networks", *International Journal of Engineering Research*, vol. 2, núm. 11, p. 6, 2013.
- [26] A. C. Egas, F. Gil-Castiñeira, E. Costa-Montenegro and J. S. Silva, "Automatic allocation of identifiers in linear wireless sensor networks using link-level processes," 2016 8th IEEE Latin-American Conference on Communications (LATINCOM), pp. 1-6, 2016.
- [27] B. R. Al-Kaseem, Y. Al-Dunainawi, y H. S. Al-Raweshidy, "End-to-End Delay Enhancement in 6LoWPAN Testbed Using Programmable Network Concepts", *IEEE Internet Things J.*, vol. 6, núm. 2, pp. 3070–3086, abr. 2019.
- [28] M. O. E. Alberto, «RED INALÁMBRICA DE SENSORES A TRAVÉS DE 6LOWPAN PARA UNA AGRICULTURA DE PRECISIÓN APLICADO EN LA HACIENDA CANANVALLE DE LA CIUDAD DE IBARRA», Quito: Pontificia Universidad Católica del Ecuador, [En línea]. Disponible en: <http://repositorio.puce.edu.ec/bitstream/handle/22000/7897/9.56.000617.pdf?sequence=4&isAllowed=y>. [Accedido: 25-mar- 2022]
- [29] "6LoWPAN demystified", Texas Instruments, p. 13, 2014. Disponible en: https://www.ti.com/lit/wp/swry013/swry013.pdf?ts=1656216795278&ref_url=https%253A%252F%252Fwww.google.com%252F. [Accedido: 04-abr- 2022]
- [30] Atmel, "ZigBit 2.4GHz Single chip Wireless Module", p. 27, 2014. Disponible en: <https://rocelec.widen.net/view/pdf/ovsxqx466n/ATMLS09552-1.pdf?t.download=true&u=5oefqw>. [Accedido: 07-abr- 2022]
- [31] "Getting Started with Atmel ZigBit Modules", 2022. Disponible en: <https://forum.digikey.com/t/getting-started-with-atmel-zigbit-modules/12893> [Accedido: 10-abr- 2022]
- [32] "ZigBit Extension User Guide (USER GUIDE)", Atmel, p. 21. Disponible en: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42186-ZigBit%20Extension-User-Guide.pdf>. [Accedido: dic. 15, 2021]
- [33] A. Asperheim, R. Sjøen, y K. Skaar, "Design and Implementation of a Rudimentary WirelessHART Network", UNIVERSITY OF OSLO: Department of Informatics, p. 249, 2012.

- [34] “Programador/depurador Atmel-ICE - Microchip | DigiKey”. Disponible en: <https://www.digikey.com/es/product-highlight/a/atmel/atmel-ice-programmer-debugger>. [Accedido: 25-mar- 2022]
- [35] “SONOFF CC2531 USB Zigbee Dongle Module Sniffer Bare Board Packet Protocol Analyzer USB Interface Dongle Capture Packet”, ITEAD STUDIO OFFICIAL. Disponible en: <https://itead.cc/product/cc2531-usb-dongle/>. [Accedido: 25-mar- 2022]
- [36] A. J. G. Robayo, “ALGORITMO PARA LA TRANSMISIÓN CONFIABLE DE PAQUETES DE DATOS EN ESTRUCTURAS LINEALES A NIVEL DE LA CAPA DE ENLACE DE DATOS SIN NECESIDAD DE USAR ACUSES DE RECIBO”, Quito: Escuela Politécnica Nacional, 2017. [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/18935/1/CD-8328.pdf>. [Accedido: 20-abr-2022]
- [37] A. E. Aguaguña Aconda, “Implementación de un algoritmo para la recuperación de fallos en una topología lineal utilizando el estándar IEEE 802.15.4”, may 2019, . [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/20292> [Accedido: 01-may-2022]
- [38] L. X. Velástegui Almeida, “Optimización del flujo de datos en un prototipo de red 6LowPAN con topología lineal”, [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/15082> [Accedido: 01-jul-2022].
- [39] C. E. Velástegui Almeida, “Caracterización de retardos de 6LowPan en redes lineales a gran escala a partir de datos experimentales obtenidos en una red escala reducida”, Quito, Escuela Politécnica Nacional, 2021.
- [40] “Atmel® Studio 7 - Developer Help”. Disponible en: <https://microchipdeveloper.com/atstudio:studio7intro>. [Accedido: 18-mar- 2022]
- [41] J. M. Londoño Jiménez, “El protocolo ZIGBEE como recurso a la tecnología en la agricultura de precisión.”, Cali, Fund. Universitaria Católica Lumen Gentium, 2016.

ANEXOS

ANEXO I. Códigos fuente desarrollados (Formato Digital)

ORDEN DE EMPASTADO