

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**REDES DE SENSORES INALÁMBRICOS PARA IOT
ALGORITMO PARA LA ELIMINACIÓN DE TRAMAS REPETIDAS EN
AMBIENTES CON TOPOLOGÍA LINEAL QUE OPERAN CON EL
PROTOCOLO IEEE 802.15.4.**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

ISIDRO MOISÉS NAVARRETE MÉNDEZ

isidro.navarrete@epn.edu.ec

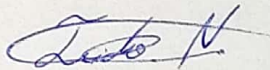
DIRECTOR: CARLOS ROBERTO EGAS ACOSTA, MSc.

carlos.egas@epn.edu.ec

DMQ, 11 de abril 2023

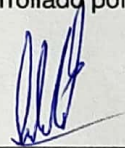
CERTIFICACIONES

Yo, ISIDRO MOISÉS NAVARRETE MENDEZ declaro que el trabajo de integración curricular 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. Yo, ISIDRO MOISÉS NAVARRETE MENDEZ declaro que el trabajo de integración curricular 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.



Isidro Moisés Navarrete Mendez

Certifico que el presente trabajo de integración curricular fue desarrollado por ISIDRO MOISÉS NAVARRETE MENDEZ, bajo mi supervisión.



Ing. Carlos Roberto Egas Acosta, MSc

DIRECTOR DE PROYECTO

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ISIDRO MOISÉS NAVARRETE MEDNEZ

CARLOS ROBERTO EGAS ACOSTA, PHD

DEDICATORIA

El presente trabajo de integración curricular esta dedicado a todas las personas que han hecho posible que pueda obtener los conocimientos, formación y valores necesarios con los cuales mi vida toma rumbo y significado.

A mi padre que ha pesar de su pronta partida supo dejarme claro todo lo antes mencionado y mucho más.

A mi madre luchadora incansable para mejorar nuestra vida.

A mis hermanos completamente dedicados a sus pasiones académicas.

A mi abuelita que nunca nos descuidó y siempre atendió con todo el amor del mundo.

A mis tíos con los que siempre pude contar.

A mis maestros y amigos, los cuales convirtieron mi vida académica en una travesía llena de agradables momentos y retos.

A mi pareja por estar ahí dándome su corazón, confianza y apoyo, en cualquier situación.

AGRADECIMIENTO

Agradezco de la manera más profunda a todas las personas que me han permitido culminar con esta etapa, y por toda la ayuda incondicional que día a día sentí por parte de mi familia y amigos.

A mi tutor Ing. Carlos Egas Acosta M.sc., por darme la oportunidad de realizar este trabajo en el cual desarrolle mis habilidades y capacidades.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	VI
RESUMEN	X
ABSTRACT	XI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	3
1.4.1 Estándar IEEE 802.15.4	3
1.4.2 Arquitectura IEEE 802.15.4	4
1.4.3 Capa Física	5
1.4.4 Mecanismos de acceso al medio CSMA-CA	9
1.4.5 Transferencia de datos	10
1.4.6 Red de Sensores Inalámbricos	11
1.4.7 Red de Sensores Inalámbricos Lineales	11
1.4.8 Componentes de una WSN	12
1.4.9 Transceptor ATZB-256RFR2-XPRO	12
2 METODOLOGÍA	15
2.1 Problemática	15
2.2 Diseño y desarrollo del algoritmo	16
2.2.1 Descripción general del algoritmo	16
2.3 Requerimientos necesarios	22
2.3.1 Hardware necesario para la implementación	22
2.3.2 Software necesario para la implementación	24
2.4 Codificación del algoritmo en el software	25
2.4.1 Inicio del proyecto	25

2.4.2	Programación del algoritmo en el software	31
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	37
3.1	Configuraciones iniciales	37
3.1.1	Elementos usados en las pruebas	37
3.1.2	Parametros iniciales en el software	38
3.1.3	Topología Lineal	39
3.2	Pruebas ejecutadas	41
3.2.1	Prueba 1: Transmisión realizada al mismo tiempo	41
3.2.2	Prueba 2: Transmisión sin el uso del algoritmo diseñado	43
3.2.3	Prueba 3: Transmisión con el uso del algoritmo diseñado y detección de ruptura por un nodo	44
3.2.4	Prueba 4: Transmisión con el uso del algoritmo diseñado y detección de ruptura por dos nodos	45
3.2.5	Prueba 5: Transmisión con el uso del algoritmo diseñado y detección de ruptura por tres nodos	46
3.2.6	Prueba 6: Transmisión con el uso del algoritmo diseñado y detección de ruptura por cuatro nodos	47
3.2.7	Prueba 7: Transmisión con el uso del algoritmo diseñado y detección de ruptura por cinco nodos	48
3.2.8	Problemas detectados	49
3.2.9	Comparación	49
3.2.10	Análisis de los resultados obtenidos	50
3.3	Conclusiones	51
3.4	Recomendaciones	52
4	REFERENCIAS BIBLIOGRÁFICAS	53
5	ANEXOS	I
5.1	ANEXO I: REPOSITORIO Y BASE DE DATOS DEL PROYECTO	I
5.2	ANEXO II: MANUAL ATMEL 256RFR2	II
5.2.1	Atmel software Framework	II
5.2.2	II. Instalación Wireless composer 6.2	IV

ÍNDICE DE FIGURAS

1.1	ESTÁNDAR IEEE 802.15.4 [5].	5
1.2	Modos de operación MAC [6].	7
1.3	Trama IEEE 802.15.4 [6].	8
1.4	Ejemplo de LWSN [7].	11
1.5	Diagrama de ATZB-256RFR2-XPRO [9].	13
2.1	Diagrama de flujo del nodo externo.	17
2.2	Topología lineal, alcance de la señal y direcciones en orden jerárquico de todos los nodos.	17
2.3	Envío y recepción mensaje “Ruptura”.	18
2.4	Envío “Ruptura T 0x0001”, “Ruptura T 0x0002” y “Ruptura T 0x0003”.	19
2.5	Envío y recepción del mensaje “Ruptura F 0x0003” proveniente de 0x0004.	19
2.6	Envío y recepción del mensaje “Ruptura F 0x0003” proveniente de 0x0006.	20
2.7	Envío y recepción del mensaje “Fin Ruptura”.	20
2.8	Funcionamiento del algoritmo y su Diagrama de flujo.	21
2.9	ATZB-256RFR2-XPRO y Alimentación.	22
2.10	Atmel-ICE Debugger kit básico [12].	23
2.11	CC2531 USB Dongle [13].	24
2.12	MicrochipStudio.	25
2.13	Nuevo proyecto en MicrochipStudio.	26
2.14	Tipo de proyecto.	26
2.15	Tipo de microcontrolador.	27
2.16	Configuración del Transceptor.	27
2.17	Configuración de transmisión y recepción.	28
2.18	Ventana de edición.	28
2.19	Ventana Tools.	29
2.20	Ventana Tools.	29
2.21	Ventana Tools.	30
2.22	Ventana Tools.	30
3.1	Diagrama del prototipo implementado.	38
3.2	Packet Sniffer primera configuración.	38
3.3	Packet Sniffer primera configuración.	39

3.4	Prototipo Implementado.	39
3.5	Secuencia de nodos.	40
3.6	Conexión a computadora.	40
3.7	Capturas de mensajes prueba 1 Sniffers.	42
3.8	Transmisión de nodos prueba 1.	42
3.9	Captura de mensajes prueba 2.	43
3.10	Transmision de nodos prueba 2.	44
3.11	Captura de mensajes prueba 3.	44
3.12	Transmisión de nodos prueba 3.	45
3.13	Captura de mensajes prueba 4.	45
3.14	Transmisión de nodos prueba 4.	46
3.15	Captura de mensajes prueba 5.	46
3.16	Transmisión de nodos prueba 5.	47
3.17	Captura de mensajes prueba 6.	47
3.18	Transmisión de nodos prueba 6.	48
3.19	Captura de mensajes prueba 7.	48
3.20	Transmisión de nodos prueba 7.	49
5.1	Paso 1.	II
5.2	Paso 2.	II
5.3	Paso 3.	III
5.4	Paso 4.	III
5.5	Paso 5.	IV
5.6	Paso 6.	IV
5.7	Paso 7.	V
5.8	Paso 8.	V
5.9	Paso 9.	VI
5.10	Paso 10.	VI

ÍNDICE DE CUADROS

1.1	Bandas de frecuencia utilizadas por 802.15.4 [5].	6
3.1	Elementos del prototipo implementado	37
3.2	Tiempos de captura prueba 1.	41
3.3	Comparación de las pruebas ejecutadas.	50

RESUMEN

IEEE 802.15.4 es un protocolo de comunicación inalámbrica de corto alcance y baja potencia que está diseñado para que dispositivos como sensores y actuadores se comuniquen dentro de una red. El protocolo en conjunto con los nodos o sensores poseen limitaciones, y la manera más eficiente de evitar un consumo excesivo de recursos es creando algoritmos que sean capaces de adaptarse a las diferentes situaciones que se presenten, como puede ser el caso de la redundancia de mensajes, el cual consume energía, tiempo y procesamiento de todos los nodos.

En el presente trabajo se desarrolla un algoritmo capaz de eliminar tramas repetidas en ambientes con topología lineal, capturando y comparando los mensajes enviados por los sensores que operan con el protocolo IEEE 802.15.4, demostrando así que es posible implementar el algoritmo sobre los nodos ATZB-256RFR2-XPRO.

El trabajo cuenta con 3 capítulos en total. En el primer capítulo se estudian los conceptos, componentes y características de la red de sensores, de la topología lineal, del protocolo IEEE 802.15.4, y del nodo ATZB-256RFR2-XPRO. En el segundo capítulo se detalla el desarrollo del algoritmo y su codificación en los nodos. Además, se desarrolla el código necesario dentro de un sensor externo para que los nodos tengan la capacidad de generar mensajes simultáneos a través de la red con topología lineal. En el tercer capítulo se muestran los resultados del algoritmo, realizando una comparación de las secuencias de mensajes recibidos y transmitidos.

PALABRAS CLAVE: IEEE 802.15.4, Topología lineal, Red de sensores inalámbricos, Algoritmo, Redundancia.

ABSTRACT

IEEE 802.15.4 is a short-range, low-power wireless communication protocol designed to facilitate the communication between devices, such as sensors and actuators, within a network. However, the protocol along with the nodes and the sensors are limited and can be inefficient. The most effective way to avoid excessive consumption of resources is to build algorithms with the potential to adapt to different situations that consume energy, time, and processing of all nodes such as message redundancy.

In this study, we present an algorithm capable of eliminating repeated frames in environments with linear topology. Moreover, the algorithm is able to capture and compare the messages sent by the sensors that operate with the IEEE 802.15.4 protocol. With this, we demonstrate that it is possible to implement the algorithm on the nodes ATZB-256RFR2-XPRO.

This project is organized into three chapters . The first chapter focuses on the concepts, components and characteristics of the sensor network, the linear topology, the IEEE 802.15.4 protocol, and the ATZB-256RFR2-XPRO node. The second chapter provides details on the development of the algorithm and its coding in the nodes. In addition, it contains the steps required to develop the code within an external sensor so that the nodes have the capacity to generate simultaneous messages through the network with linear topology. In the third chapter the results of the algorithm are shown, making a comparison of the sequences of messages received and transmitted.

KEY WORDS: IEEE 802.15.4, Linear topology, Wireless sensor network, Algorithm, Redundancy.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el mundo actual referente a las redes de sensores inalámbricos (WSN: Wireless Sensor Networks) son una tendencia tecnológica, la cual se encuentra en constante desarrollo de nuevas aplicaciones, son consideradas una opción viable para cualquier sistema de monitoreo que requiera obtener datos provenientes de múltiples sensores ubicados en un área de interés, su principal característica es la versatilidad de configuración [1].

Muchas aplicaciones especialmente industriales requieren utilizar e implementar redes de sensores inalámbricos lineales (LWSN: Lineal Wireless Sensor Networks), estas hacen uso de una topología lineal que presenta un rápido despliegue, flexibilidad y bajo costo en sistemas confiables, se caracterizan por el uso de un número óptimo de nodos ubicados a lo largo de una infraestructura lineal, los cuales mantienen una posición fija a lo largo del tiempo, su principal restricción radica en los pocos nodos vecinos que posee el sistema, ya que limita las rutas de transmisión factibles, lo que causa mayor probabilidad de pérdida de datos, además no es posible utilizar soluciones tradicionales provenientes de WSN, debido a que el uso de protocolos de enrutamiento complejos perjudican al rendimiento en una topología lineal [2].

Los sensores principalmente utilizan el modo de comunicación de muchos a uno, es decir, muchos nodos transmiten lecturas de sensores o eventos de red a una estación base central por medio de una topología de enlace de varios pasos. Sin embargo, los nodos vecinos en la WSN a menudo ven los mismos eventos, reaccionando a ellos y enviando paquetes a la estación base, esto genera un exceso de tráfico que desperdicia energía y ocupa espacio en la red, generando una gran pérdida si la información tiene que ser precisa, rápida y óptima para un correcto funcionamiento del sistema. Con el fin de reducir los mensajes redundantes las redes de sensores utilizan técnicas de procesamiento de redes como la agregación y eliminación de datos duplicados, este procesamiento en la red requiere que los nodos intermedios puedan acceder, modificar o eliminar el contenido de los mensajes enviados por otros nodos [2].

El Trabajo de Integración Curricular, muestra el funcionamiento e implementación del algoritmo capaz de eliminar la redundancia de información en los nodos que poseen una topología lineal. Se hace uso del protocolo IEEE 802.15.4.

1.1 OBJETIVO GENERAL

Implementar un algoritmo para la eliminación de tramas repetidas en ambientes con topología lineal que operan con el protocolo IEEE 802.15.4.

1.2 OBJETIVOS ESPECIFICOS

- Estudiar las características del estándar IEEE 802.15.4 para el desarrollo del algoritmo.
- Describir las características de redes inalámbricas de sensores con topologías lineal multisalto.
- Estudiar el funcionamiento del microcontrolador ATmega256RFR2.
- Diseñar un algoritmo que permita la detección tramas redundantes en los nodos a nivel de capa de enlace, basado en el estándar IEEE 802.15.4.
- Implementar un prototipo para verificar el funcionamiento del algoritmo.
- Analizar los resultados obtenidos en base a las respuestas de los nodos.

1.3 ALCANCE

Para obtener la solución se propone diseñar un algoritmo estudiando al estándar IEEE 802.15.4, permitiendo que sea posible manipular los sensores, eliminando las tramas repetidas provenientes de varios nodos que captan eventos similares, es decir información redundante en el sistema. La metodología por utilizar es la cuantitativa, por lo cual es fundamental la recolección de la información y comparación entre el sistema con y sin el algoritmo [2].

El algoritmo funcionara dentro de una WSN con topología lineal multisalto, por lo que es necesario describir las características de estas redes, utilizadas de manera industrial, esto

permitirá una mejor comprensión y explicación de su funcionamiento.

En una red inalámbrica cuando un nodo transmite una trama, a partir de ese momento, todos los nodos activos en el área de cobertura la reciben; además, en una topología lineal, el enrutamiento es muy simple, ya que los paquetes solo pueden ir en una de las dos direcciones de la infraestructura lineal y los nodos solo tienen que decidir si tienen que retransmitir un paquete o no. Por lo tanto, el funcionamiento que normalmente está implementando dentro de una capa separada se puede integrar dentro de las funcionalidades de la capa 2 (enlace) [3].

En la creación del algoritmo diseñado se utiliza el lenguaje de programación C/C++, el cual será utilizado para el uso del microcontrolador ATmega256RFR2, el mismo que será combinado con un transceptor de 2,4 GHz [4].

Se implementará una red prototipo WSN con topología lineal, la cual estará conformado por 6 nodos, en los cuales se realizarán varias pruebas del funcionamiento del algoritmo diseñado, de esta forma se obtendrán los datos necesarios para su posterior análisis.

Finalmente se realizará un análisis de las respuestas obtenidas, llegando a generar conclusiones y recomendaciones, las cuales permitirán determinar el uso y capacidad del algoritmo planteado.

1.4 MARCO TEÓRICO

1.4.1 Estándar IEEE 802.15.4

IEEE 802.15.4 fue diseñado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE: The Institute of Electrical and Electronics Engineers) con el fin de utilizarlo en redes inalámbricas con una pequeña tasa de transmisión y recepción (LR-WPAN: Low Rate Wireless Personal Area Network), entre sus características principales cabe mencionar el bajo consumo de recursos mediante celdas de energía o baterías, representando mucho tiempo de operación del dispositivo; además, su transmisión es confiable debido a que utiliza mecanismos como el acceso al medio por CSMA/CA, mensajes de confirmación ACK, retransmisiones y manteniendo una arquitectura básica la cual posee una capa física y una de control [5].

IEEE 802.15.4 posee dos tipos completamente diferentes de dispositivos:

- **Dispositivos de funcionalidad completa (FFD: Full Function Device):** Estos dispositivos operan en tres modos; El Coordinador encargado de realizar la sincronización; Coordinador de red de área personal (PAN: Personal Area Network) el cual asocia dispositivos y Dispositivo simple ubicado dentro de la PAN [5].
- **Dispositivos de funcionalidad reducida (RFD: Reduced Function Device):** Capaces de funcionar con una muy simple infraestructura e implementación, cabe mencionar que no puede ser coordinador PAN y es descrito como un dispositivo final ya que no le es posible transmitir cantidades de información de considerable tamaño y puede asociarse a un solo FFD [5].

1.4.2 Arquitectura IEEE 802.15.4

IEEE 802.15.4 especifica un conjunto de subcapas y capas, desplegadas en la Figura 1.1, estas deben cumplir con el parámetro de interconexión de un sistema abierto (OSI: Open System Interconnection). Un dispositivo que maneje IEEE 802.15.4 consta de una parte física (PHY: physical layer) y una subcapa de manejo de acceso al medio (MAC: Media Access Control), la capa PHY incluye un transceptor de radiofrecuencia mientras que la subcapa MAC facilita el acceso al canal físico. IEEE 802.15.4 al ser ingresado por la IEEE, cumplió con el objetivo de complementar necesidades y requerimientos que anteriormente no fueron cubiertos y los principales son:

- IEEE 802.15.1 Bluetooth: Tecnología proveniente de la red inalámbrica de velocidad lenta y potencia baja, utilizado en comunicaciones punto a punto.
- IEEE 802.15.3 High-Rate WPAN (WPAN: Wireless Personal Area Network): Fue desarrollado para ser utilizado en aplicaciones con una gran cobertura territorial y una alta tasa de información, aunque cabe mencionar que no existen muchas aplicaciones que requieran altas características.

El estándar por analizar fue desarrollado para crear y manejar protocolos óptimos que sean sencillos y maniobrables, su implementación se caracteriza por tener bajo uso de recursos como la energía; además de una arquitectura sencilla, tasas de transmisión bajas, colisiones evitadas con CSMA/CA, aplicaciones en tiempo simultaneo, soporte integrado con la finalidad de que la comunicación sea más segura y de presupuesto bajo [5].

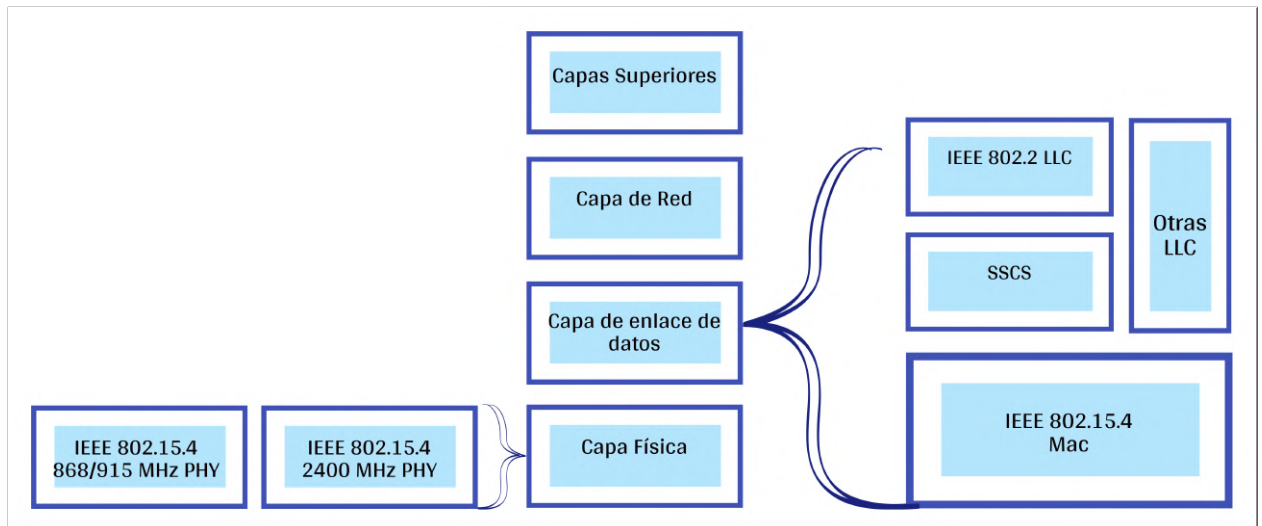


Figura 1.1: ESTÁNDAR IEEE 802.15.4 [5].

1.4.3 Capa Física

La capa física (PHY: Physical Layer) es responsable de interactuar con el medio de transmisión físico, intercambiando información con sí mismo y con una capa superior (capa MAC), para realizar esto debe emplear un canal de radio y una técnica de modulación determinada. Sus responsabilidades en relación con el medio físico radioeléctrico son las siguientes [5]:

- Activación y apagado del transceptor que utiliza el canal de radio.
- Detección de energía (ED- Energy detection).
- Parámetro de calidad de enlace (LQI-Link Quality Indication).
- Evaluación del canal transparente (CCA- Clear Channel Assessment).

Para describir las características indicadas hay que mencionar las bandas de frecuencia especificadas por la capa física de IEEE 802.15.4.

La capa física de IEEE 802.15.4 posee tres bandas de frecuencia operativas: Una de 2,4 GHz, otra de 915 MHz y por último una de 868 MHz. Cabe mencionar que en medio de 868 y 868,6 MHz se encuentra solo un canal, entre 902 y 928 MHz hay 10 canales y entre 2,4 y 2,4835 GHz hay 16 canales [5].

A 2,4 GHz la velocidad de datos es de 250 kbps, a 915 MHz la velocidad es de 40 kbps y a 868 MHz la velocidad es 20 kbps; además, cabe mencionar, que debido a la disminución de

las pérdidas de propagación las frecuencias más bajas son mejores para rangos de transmisión más largos. Las transmisiones de baja velocidad tienen un área de cobertura mayor y una sensibilidad mejorada, es decir, cuanto mayor sea la tasa, mayor será el rendimiento y menor el ciclo de trabajo, así como también la latencia.

Las bandas de frecuencia utilizan una técnica de espectro ensanchado por secuencia directa (DSSS-Direct Sequence Spread Spectrum) para la transmisión [5]. A continuación. Dentro de la Tabla 1.1 se resume a cada banda de frecuencia.

Tabla 1.1: Bandas de frecuencia utilizadas por 802.15.4 [5].

Bandas	N° de Canales	Técnica Ensanchamiento	Modulación	Velocidad símbolo	Tasa de bits
868 MHz	1	Binary DSSS	BPSK	20	20
915 MHz	10	Binary DSSS	BPSK	40	40
2.4 GHz	16	16-array DSSS	O-QPSK	62.5	250

1.4.3.1 Subcapa MAC

La subcapa MAC proveniente de IEEE 802.15.4 cumple con la función de ser enlace entre las capas superiores y la capa física; los servicios que podemos encontrar en la subcapa son:

- El servicio de datos
- El Servicio de administración

El primer servicio hace que la transmisión de información entre capas se posible, y el segundo servicio se encarga de administrar los medios para que le sea posible acceder a una capa superior a la subcapa [6].

El protocolo de acceso al canal que se utiliza dentro de la capa MAC es conocido como de acceso múltiple por detección de portadora y prevención de colisiones (CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance), el cual maneja el modo beacon activado o desactivado.

Cuando el beacon esta activado se hace uso de la estructura de la supertrama con el fin de manejar una buena comunicación dentro de los nodos asociados a la red de área personal (PAN: Personal Area Network), por lo que el coordinador crea una supertrama y la

transmite repetidamente en un cierto periodo de tiempo a los demás nodos, utilizando cada trama beacon. Una supertrama consta de 16 ranuras con igual tamaño, seguidamente de un periodo de inactividad; además se encuentra limitada por dos tramas beacon consecutivas y por periodos de acceso o libre contención [6].

En el modo beacon desactivado no se emplea supertramas, el encargado de proporcionar el control de acceso al medio es el mecanismo CSMA/CA no ranurado, es decir las tramas transmitidas pueden ser enviados con acuse de recibo o no y se deben enviar según la norma [6]. Dentro de la Figura 1.2 se encuentra los modos de operación.

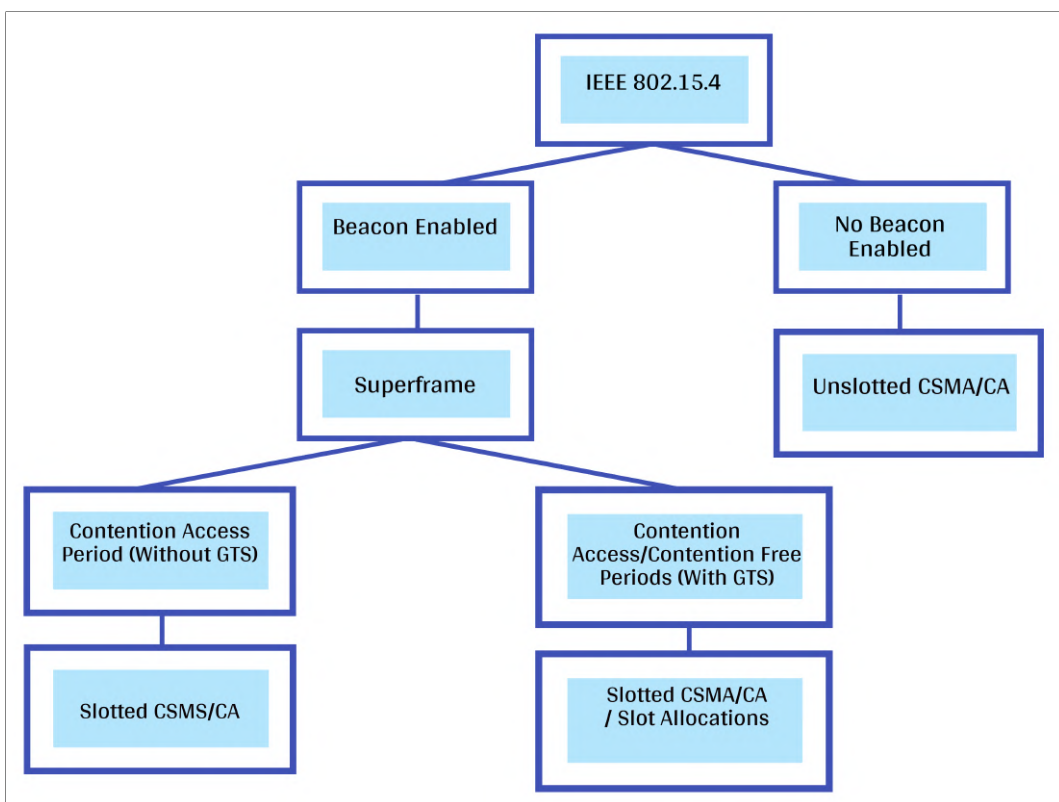


Figura 1.2: Modos de operación MAC [6].

1.4.3.2 Estructura de las tramas MAC

Dentro de la trama MAC IEEE 802.15.4 existe una magnitud de 127 bytes como máximo; además, esta se puede dividir en Encabezado MAC, Unidad de Datos de Servicio MAC y Pie de Página MAC (MHR MAC Header, MSDU MAC Service Data Unit y MFR MAC Footer). Dentro de la Figura 1.4 está la trama IEEE 802.15.4, con todas las partes mencionadas, y las cuales poseen los siguientes campos [6]:

- **MAC Header (MHR):** Control de Trama, Número de secuencia, Identificador PAN destino, Dirección de destino, Identificador PAN origen, Dirección de origen.
- **MAC Service Data Unit (MSDU):** Carga útil.
- **MAC Footer (MFR):** Pasos de comprobación de fotogramas (FCS Frame Check Sequence)

A continuación, se detallan los campos mencionados anteriormente:

- **Control de Trama:** Define la clase de trama que es, posee banderas para el control. Su magnitud es de 16 bits.
- **Número de secuencia:** Identificador de pasos o secuencia único de cada trama, en ACK se determina si una comunicación fue exitosa. Su longitud es de 8 bits.
- **Identificador PAN destino:** Identificador único de red PAN proveniente del receptor de la trama. Su longitud es de 16 bits.
- **Dirección de destino:** Dirección expresada en hexadecimal del receptor de la trama. Su longitud es de 16 o 64 bits.
- **Identificador PAN origen:** Identificador único de red PAN proveniente del transmisor de la trama, su longitud es de 16 bits.
- **Dirección de origen:** Dirección del transmisor de la trama. Su longitud es de 16 o 64 bits.
- **Carga útil:** Información detallada que varía según el tipo de trama IEEE 802.15.4. De longitud variable, hay que tener en cuenta el tamaño máximo posible.
- **FCS:** Incluye a la verificación de redundancia cíclica (CRC cyclic redundancy check). Su longitud es de 16 bits.

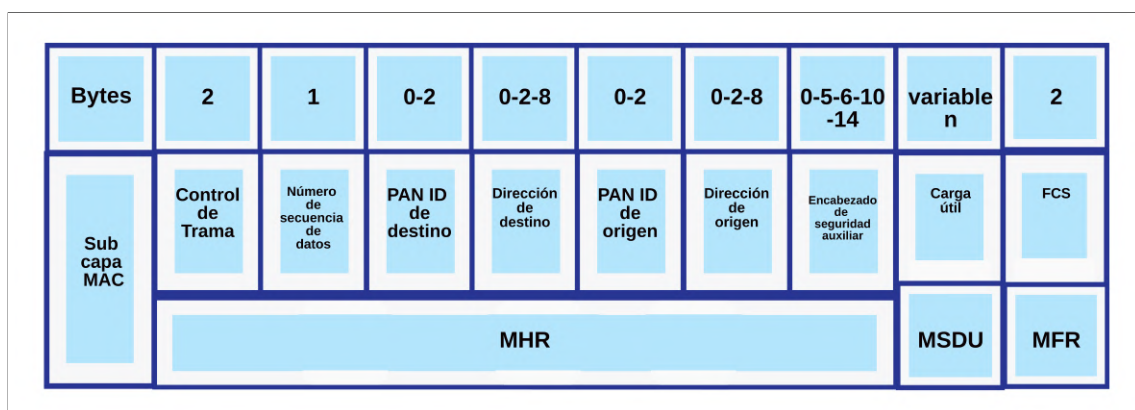


Figura 1.3: Trama IEEE 802.15.4 [6].

1.4.4 Mecanismos de acceso al medio CSMA-CA

Existen dos mecanismos de acceso al medio CSMA/CA que se encuentran en el estándar IEEE 802.15.4:

- CSMA/CA ranurada, utilizada en el modo beacon.
- CSMA/CA no ranurada, utilizada en el modo no beacon.

En los dos mecanismos, se está utilizando el algoritmo de backoff, cabe mencionar que un periodo de backoff equivale a 20 símbolos, estos son tomados como valor de unidad básica de tiempo descrita por el protocolo MAC [6]. CSMA/CA utiliza las siguientes variables con la finalidad de configurar el acceso al medio:

- **Exponente de back backoff (BE: Backoff Exponent):** Es usado para calcular el periodo de backoff que se espera un nodo antes de analizar y determinar si existe un canal libre o no, BE depende del tipo de modulación y la frecuencia [6].
- **La ventana de contención (CW: Contention Window):** Se la define como una cantidad de periodos de backoff para asegurar que se libere el canal, CW utilizado únicamente en el mecanismo ranurado de CSMA/CA [6].
- **Número de backoffs (NB: Number of Backoff):** Es una determinada cantidad de veces que CSMA/CA trata de acceder al canal [6].

1.4.4.1 Mecanismo CSMA/CA ranurado

Este mecanismo es empleado en las redes beacon, dentro de este la supertrama se está alineando con los límites de los periodos de backoff de los nodos ubicados dentro de la red PAN, cabe mencionar que el coordinador comienza enviando la trama beacon permitiendo que se alinee con el principio del primer periodo de backoff [6].

Dentro de una red beacon en el momento que se transmite una trama el nodo inicia el algoritmo de backoff tomando en cuenta que debe hacerlo antes de transmitir la supertrama; también, verifica si el canal se encuentra ocupado, para poder iniciar la transmisión en el posterior slot de tiempo accesible, si el canal está ocupado se inicia el algoritmo de backoff antes de intentar transmitir, en este periodo se transmite el beacon y las tramas de acuse de recibo sin utilizar CSMA/CA [6]

1.4.4.2 Mecanismo CSMA/CA no ranurado

Es utilizado en redes con beacon desactivado; además, para que un nodo quiera transmitir primeramente debe haber ejecutado el algoritmo de backoff, cuando termina este periodo de tiempo y el canal se encuentra libre, el nodo puede comenzar con la transmisión, si no se encuentra el canal libre nuevamente ejecuta el algoritmo de backoff; además las tramas de acuse de recibo se envían sin emplear CSMA/CA, cabe mencionar que el presente proyecto trabaja con este mecanismo [6].

1.4.5 Transferencia de datos

IEEE 802.15.4 consta de tres formas en las cuales se puede realizar la transferencia de datos utilizando dispositivos que deben ser configurados con varias funciones como coordinador o nodos intermedios, descritas a continuación:

- **Nodo a Coordinador:** Esta transferencia de datos puede utilizar tanto CSMA/CA ranurado como no ranurado. En el primer caso después de usar la supertrama con la finalidad de sincronizar los nodos y el canal se encuentra libre para la transmisión, el nodo transmite la información al nodo coordinador y este usa el acuse de recibo para confirmar la recepción, si se utiliza el mecanismo no ranurado simplemente se envía la información y de igual forma puede hacer uso del acuse de recibo [6].
- **Coordinador a nodo:** Puede utilizar tanto CSMA/CA ranurado como no ranurado, en el primer caso el nodo coordinador transmite un mensaje en el cual menciona que tiene datos en espera de ser enviados, mientras que los nodos están censando periódicamente, cuando un nodo desee recibir los datos envía un comando MAC pidiendo una trama acuse de recibo al coordinador, cuando utiliza el mecanismo no ranurado se guarda la información hasta que se detecte que un nodo envió su trama solicitando la transferencia, también se utiliza el acuse de recibo para la confirmación de solicitud y otro acuse para confirmar la recepción [6].
- **Transferencia punto a punto:** Dentro de la red los nodos pueden comunicarse entre sí pero únicamente con los sensores que se encuentren dentro de un rango de cobertura, con la finalidad de obtener una transmisión de datos confiable los nodos intercambian información constantemente; además, pueden sincronizarse entre ellos, transmiten usando

CSMA/CA no ranurado [6].

En el presente trabajo se utiliza el tipo de transferencia de punto a punto utilizando CSMA/CA no ranurado, debido a que se debe analizar y eliminar la redundancia de los mensajes que pueden ser enviados por varios nodos al mismo tiempo, estos mensajes serán recibidos por los sensores que se ubican dentro del rango de cobertura y los cuales procesarán la información y decidirán si retransmitir o no los datos.

1.4.6 Red de Sensores Inalámbricos

Una red de sensores inalámbricos (WSN: Wireless Sensor Networks), se lo puede definir como un grupo de diferentes sensores independientes que mantienen una comunicación entre ellos dirigiendo una variedad de condiciones en diferentes puntos, su finalidad es la de recibir o transmitir datos de manera inalámbrica en una red con la capacidad de aguardar la información para ser monitoreada [7].

1.4.7 Red de Sensores Inalámbricos Lineales

Las redes de sensores inalámbricos Lineales (LWSN: Lineal Wireless Sensor Networks), son caracterizadas debido a un despliegue de sensores o nodos escasos a lo largo de la infraestructura lineal Figura 1.1. estos sensores o nodos no pueden moverse después de implementarse. Una de las principales limitaciones de la topología lineal es el número reducido de nodos vecinos que limita las rutas de transmisión factibles, tales limitaciones hacen que las soluciones tradicionales para WSNs no sean aplicables a LWSNs [7].

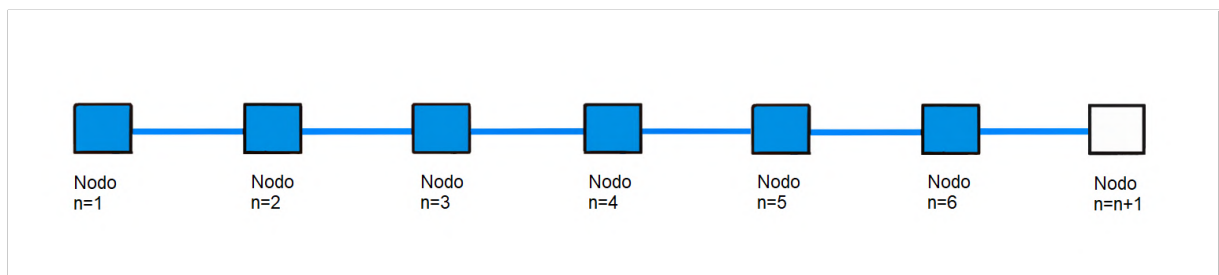


Figura 1.4: Ejemplo de LWSN [7].

1.4.8 Componentes de una WSN

Una red de sensores inalámbricos que es utilizada en la industria o en investigación está compuesta por una serie de elementos que cumplen con varias funciones indispensables para una correcta comunicación, cada elemento se programa según los parámetros necesarios de cada aplicación para posteriormente ser descargada su información a través de una conexión serial, en el documento actual se utiliza una LWSN en la cual se analiza un tramo de comunicación, es decir, varios nodos intermedios, todos los elementos básicos utilizados en una WSN y LWSN son mostrados a continuación [8]:

- **Sensores:** Estos equipos poseen diferentes tecnologías y se los puede encontrar en una gran variedad de tipos según su uso, su funcionamiento radica en tomar datos del entorno para posteriormente transformarlos en señales eléctricas.
- **Nodo Sensor:** Es considerado un procesador de radio, el cual guarda datos del sensor utilizando puertos de información, y envía los datos recolectados para posteriormente ser almacenados en un equipo principal.
- **Gateway:** Su característica principal es la de interconectar a la red de sensores inalámbricos a una red exterior o con una red de sensores diferentes.
- **Estación base:** Recolecta, guarda y muestra los datos mediante una interfaz gráfica.
- **Red inalámbrica:** Se basa en IEEE 802.15.4 o ZigBee; también, existen estándares basados por los propios fabricantes.

1.4.9 Transceptor ATZB-256RFR2-XPRO

El transceptor ATZB-256RFR2-XPRO posee un microcontrolador ATmega256RFR2 el cual permite el desarrollo de programas y aplicaciones IEEE 802.15.4., una memoria que almacena un número de serie y una dirección MAC llamada ID EEPROM10, dos pulsadores, 3 diodos LED, una antena, un conector JTAG, una cabecera con 20 entradas, 24 pines de entrada y salida, Figura 1.5 [9].

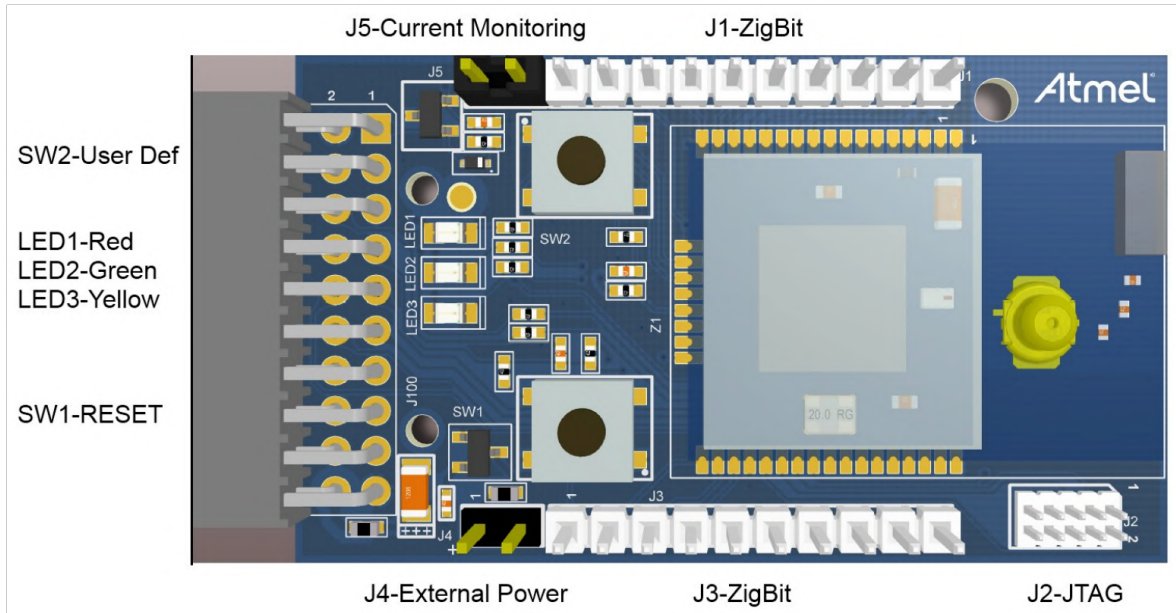


Figura 1.5: Diagrama de ATZB-256RFR2-XPRO [9].

1.4.9.1 ATmega256RFR2

El microcontrolador es de tipo RISC, es decir permite reducir el tiempo de ejecución mediante el uso de reglas cortas y simples. Cuenta con un almacenamiento de programación flash no volátil de 256 KB y un almacenamiento SRAM3 de 32 KB con el fin de operar al nodo como un dispositivo FFD [9].

Además admite un conjunto de dispositivos periféricos, lo cual permite al dispositivo ser un coordinador para la red de área personal (PAN: Personal Area Network). Por lo que el nodo puede operar como un dispositivo de funcionalidad completa o reducida. También, utiliza un oscilador interno RC9 sintonizado a 16 MHz y pre escalado a 8 MHz, por lo que está configurado para funcionar a 16 MHz con voltajes de suministro que van desde 1,8 V a 3,6 V [9].

1.4.9.2 Características de ATZB-256RFR2-XPRO

ATZB-256RFR2-XPRO posee una gran variedad de características las cuales permiten que sea un nodo adaptable a cualquier tipo de aplicación.

A continuación, las características más importantes de ATZB-256RFR2-XPRO [9]:

- Comunicación half dúplex.
- El hardware posee con especificaciones del estándar IEEE 802.15.4.
- Evaluación de canal libre (CCA).
- Detección de energía con (ED).
- Detección de inicio de trama (SFD-PHY).
- Verificación de redundancia cíclica (CRC 16).
- Acuses de recibo (ACK).
- Medición de intensidad de la señal recibida (RSSI11).
- Cálculo de la calidad de enlace (LQI).
- Control de transmisión, recepción, y retransmisión de tramas.
- Método de acceso al medio CSMA-CA.
- Maneja un amplificador de alta potencia para transmitir datos.
- La Velocidad de la información es de: 250 Kbps, 500 Kbps, 1 Mbps y 2 Mbps.
- Sensibilidad para la recepción de -100 dBm.
- Potencia de transmisión reprogramable (-17 dBm a 3.5 dBm).
- Sintetizador PLL con 5 MHz que posee un espacio dentro de canales para la banda de 2,4 GHz.
- Búfer de 128 bytes para guardar tramas recibidas o que están por transmitirse.

2 METODOLOGÍA

En este capítulo presente se explica la problemática y el algoritmo codificado para cumplir con los objetivos establecidos; además, del funcionamiento de cada parte del algoritmo que permite transmitir información para realizar la detección de las tramas repetidas y eliminar la redundancia. Posteriormente, se detalla el desarrollo e implementación del algoritmo diseñado, con el uso de programación en lenguaje C, explicando los códigos que actúan según el funcionamiento del algoritmo para la correcta ejecución del prototipo.

2.1 PROBLEMÁTICA

El crecimiento acelerado del uso de redes inalámbricas lineales en campos industriales y comerciales requiere la investigación y mejora de las redes de sensores, por ejemplo, es imposible implementar el protocolo de árbol de expansión (STP: Spanning Tree Protocol) para evitar la redundancia, porque este protocolo está diseñado para evitar bucles de red en topologías con múltiples caminos. En una topología lineal, no hay caminos alternativos para que el protocolo STP pueda evitar un bucle de red. Sin los caminos alternativos, el protocolo STP no puede funcionar [10].

La eliminación de la redundancia en topologías lineales es importante para maximizar el uso de los recursos de red y mantener una comunicación eficiente. Esto ayuda a mejorar el rendimiento de cada nodo y la red en general, al evitar que los mensajes se envíen por los mismos segmentos de la red. Además, al reducir la redundancia, se reduce el consumo de ancho de banda. También hay menos tráfico en la red y los enlaces se mantienen más estables. En general, la eliminación de la redundancia en topologías lineales ayuda a mejorar la calidad y la fiabilidad de la red.

2.2 DISEÑO Y DESARROLLO DEL ALGORITMO

2.2.1 Descripción general del algoritmo

El propósito del presente trabajo es plantear un algoritmo capaz de solucionar y afrontar los problemas de transmisión durante una emergencia; es decir, la detección de una anomalía o ruptura, en donde, varios nodos que operan dentro de una red de sensores inalámbricos con topología lineal envían de forma simultánea varias tramas con la misma información; además, el algoritmo se implementara en un prototipo, el cual se basa en el concepto de comunicación entre nodos.

En la red de sensores inalámbricos Lineales LWSN, lo primero que se toma en consideración son varios aspectos predeterminados como: la dirección o ID de cada nodo, la red PAN a la que pertenecen, el sentido en el cual se enviaran los mensajes y el evento que desencadena la activación del algoritmo.

Después de agregar la programación necesaria para recibir y retransmitir los mensajes; además, de definir los nombres de identificación, tanto de los nodos, como de la red, ya es posible comenzar con una prueba de funcionamiento, en donde uno o varios nodos intermedios detectan la anomalía, la cual podría ser como una ruptura de una tubería, lo que generaría varias transmisiones al mismo tiempo.

Para crear la anomalía o ruptura, se decidió configurar un nodo externo, el cual enviara dos señales broadcast con un tiempo de separación entre ellas de cinco segundos, cada señal incluye un mensaje (“Ruptura” y “Fin Ruptura”) respectivamente, permitiendo así, simular el mensaje de alerta o emergencia que posteriormente será detectado por varios nodos y casi al mismo tiempo, esto genera varios mensajes de alerta simultáneos, creando así redundancia.

Cabe mencionar que el estudio del origen de la anomalía no es parte de este trabajo.

La Figura 2.1 revela el diagrama de bloques del funcionamiento del nodo externo, antes mencionado.

En el momento que un mensaje es enviado de forma simultánea por varios nodos, todos son capaces de saber su procedencia, ya que independientemente de si detectaron la anomalía o no, estos saben que nodos se encuentran a sus extremos.

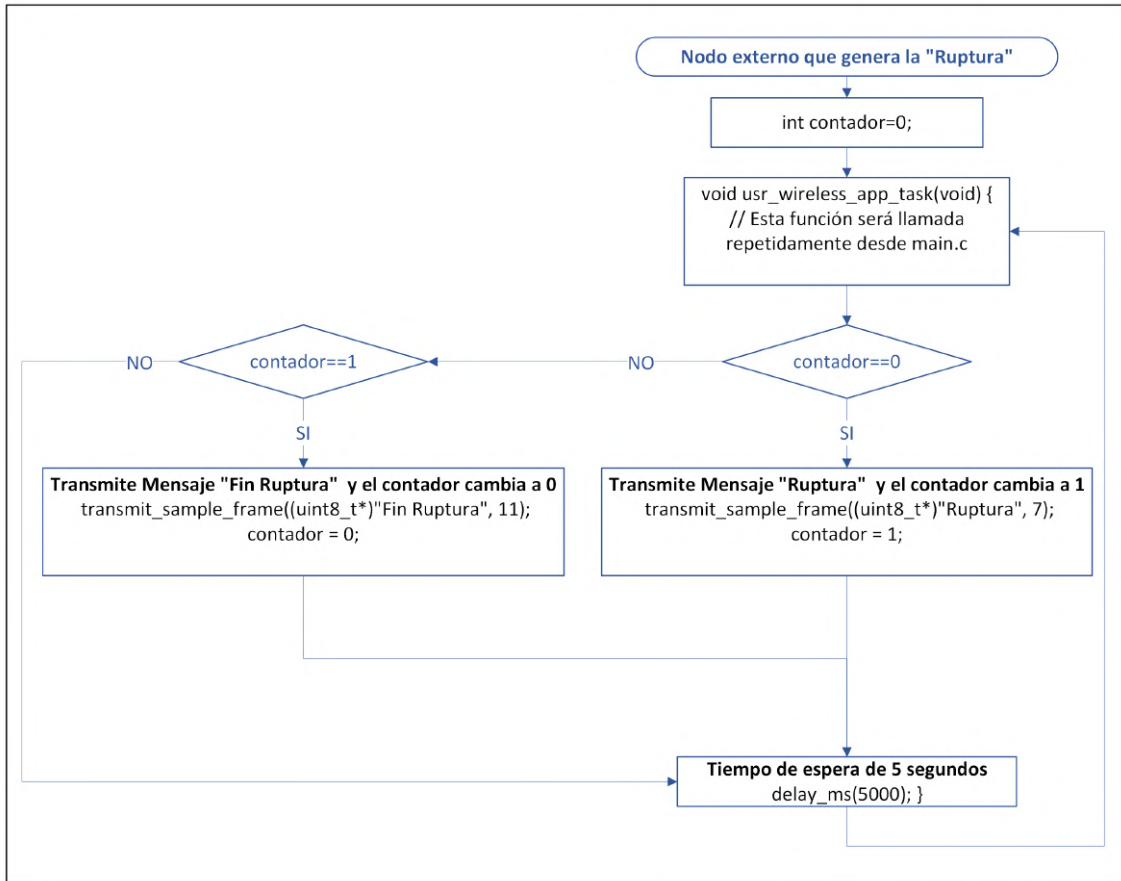


Figura 2.1: Diagrama de flujo del nodo externo.

Con fines prácticos, se decidió simular un alcance de señal que abarca la distancia de dos nodos, esto permite que la señal llegue mínimo a dos nodos, creando una situación en la cual existen varias señales y el algoritmo decidirá solo retransmitir una. La Figura 2.2 contiene la ubicación y alcance de todos los nodos, así como sus nombres.

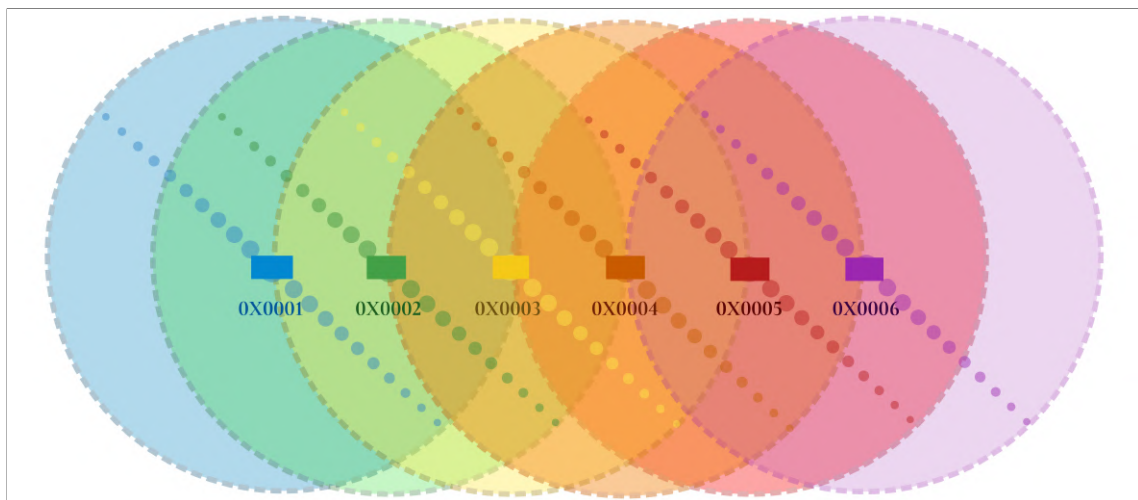


Figura 2.2: Topología lineal, alcance de la señal y direcciones en orden jerárquico de todos los nodos.

Inicialmente, todos los nodos están en modo recepción, por lo que siempre volverán a este estado después de detectar, clasificar y reenviar un mensaje.

El algoritmo comienza cuando varios nodos reciben el mensaje de "Ruptura", proveniente del nodo 0x1000, el cual es el nodo externo antes mencionado, en este primer ejemplo, esta señal llega a tres nodos y como se mencionó anteriormente la señal llega casi al mismo tiempo Figura 2.3.

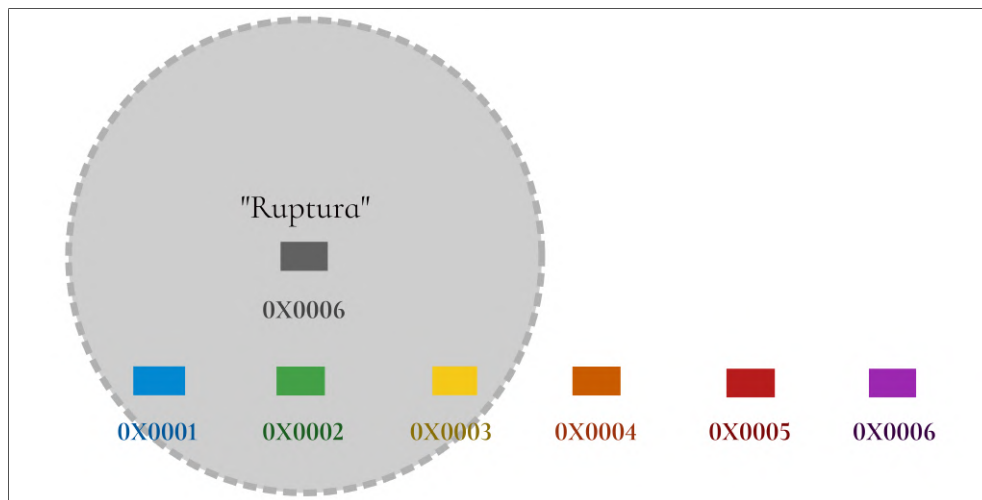


Figura 2.3: Envío y recepción mensaje "Ruptura".

A continuación, los nodos ejecutan las siguientes acciones:

- Reenviar el mensaje, en el cual se agrega un carácter identificador "T" para diferenciar si el mensaje proviene de un nodo que detectó o no la ruptura; también se agrega su ID para saber de dónde proviene la ruptura.
- Modificar una variable interna llamada "sensado": esta variable indica que el nodo ha detectado o censado la ruptura.

Los mensajes enviados anteriormente, son recibidos por los nodos adyacentes y por los nodos que no han detectado la ruptura Figura 2.3.

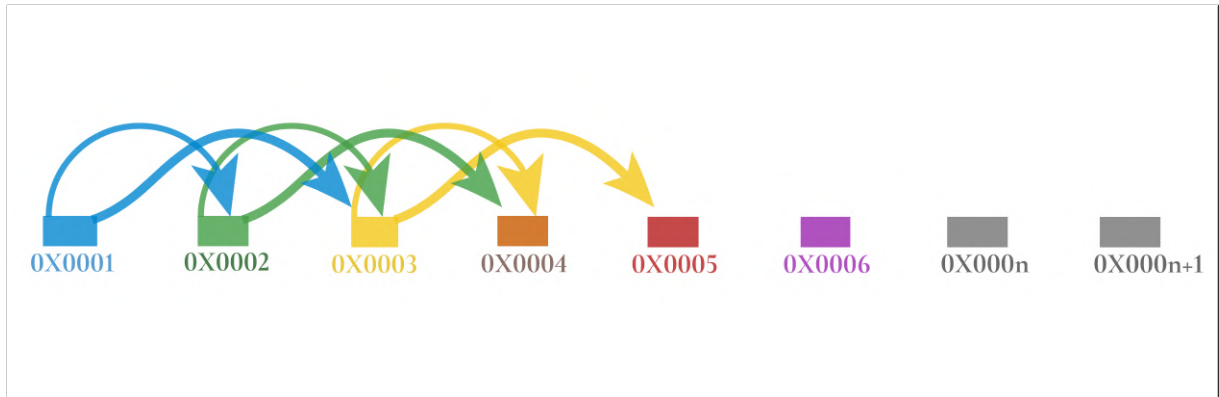


Figura 2.4: Envío “Ruptura T 0x0001”, “Ruptura T 0x0002” y “Ruptura T 0x0003”.

Estos nodos proceden a comparar varios aspectos importantes, como: la dirección de origen del mensaje, el contenido del mensaje, y su variable de detección.

Se presentan los siguientes casos posibles:

- El nodo recibe el mensaje “Ruptura T”, su variable interna indica que no ha detectado la ruptura y el mensaje proviene del nodo anterior, si estas condiciones se cumplen, el nodo se encontraría ubicado junto al último nodo que detecto la ruptura, por este motivo, se cambia la variable “T” a “F” dentro del mensaje y lo reenvía Figura 2.5.

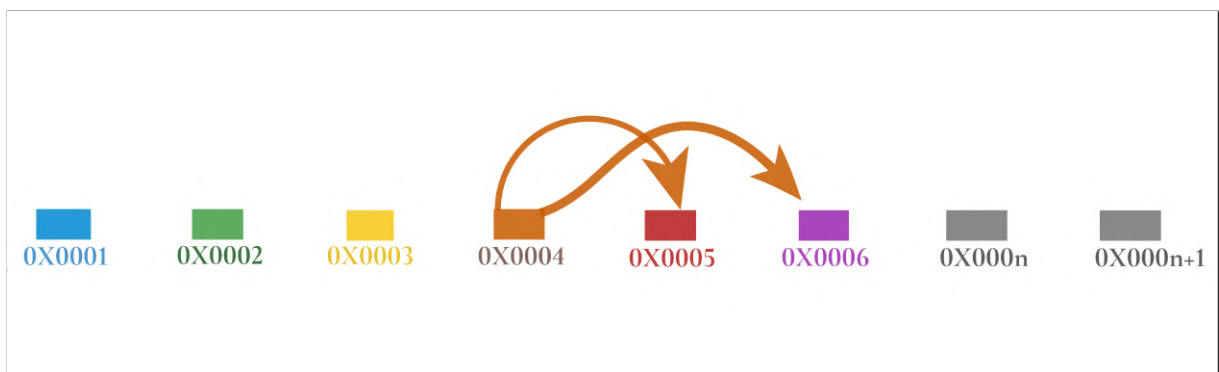


Figura 2.5: Envío y recepción del mensaje “Ruptura F 0x0003” proveniente de 0x0004.

- El nodo recibe el mensaje “Ruptura F”, su variable interna indica que no ha detectado la ruptura y el mensaje proviene del nodo que se encuentra dos posiciones por detrás, estas condiciones indican que el mensaje ya ha sido reenviado por un nodo que no ha detectado la ruptura y procede a reenviar el mensaje Figura 2.6; además, el nodo anterior no envió el mensaje ya que si esto ocurre, existirían dos mensajes iguales.

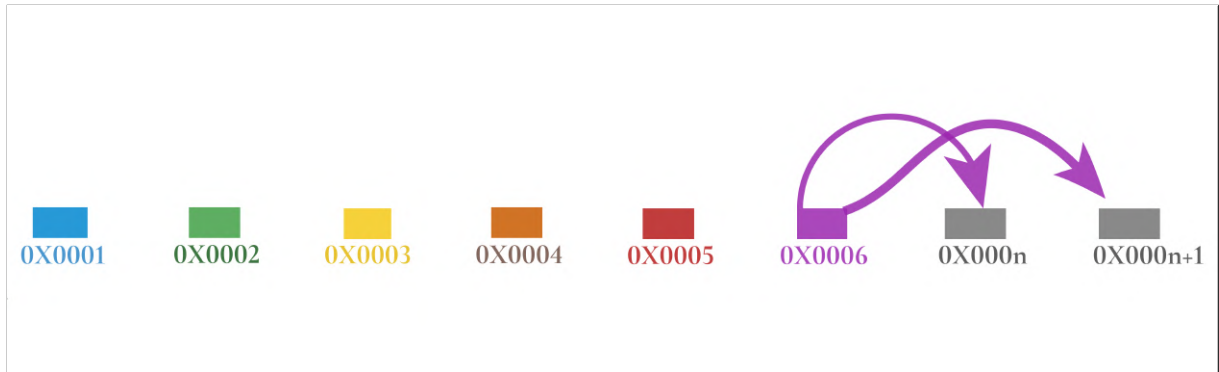


Figura 2.6: Envío y recepción del mensaje “Ruptura F 0x0003” proveniente de 0x0006.

Finalmente, los nodos que detectaron la ruptura, reciben el mensaje “Fin Ruptura”, como se muestra en la Figura 2.7, con este mensaje los nodos regresan su variable de detección al valor por defecto, con el propósito de recibir una nueva anomalía o ruptura.

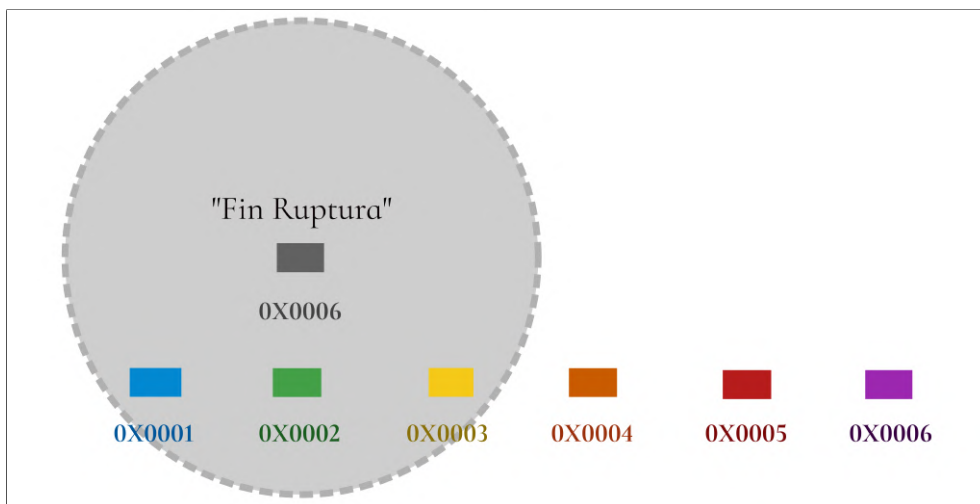


Figura 2.7: Envío y recepción del mensaje “Fin Ruptura”.

La Figura 2.8 indica con el diagrama de bloques cómo funciona el algoritmo de forma general.

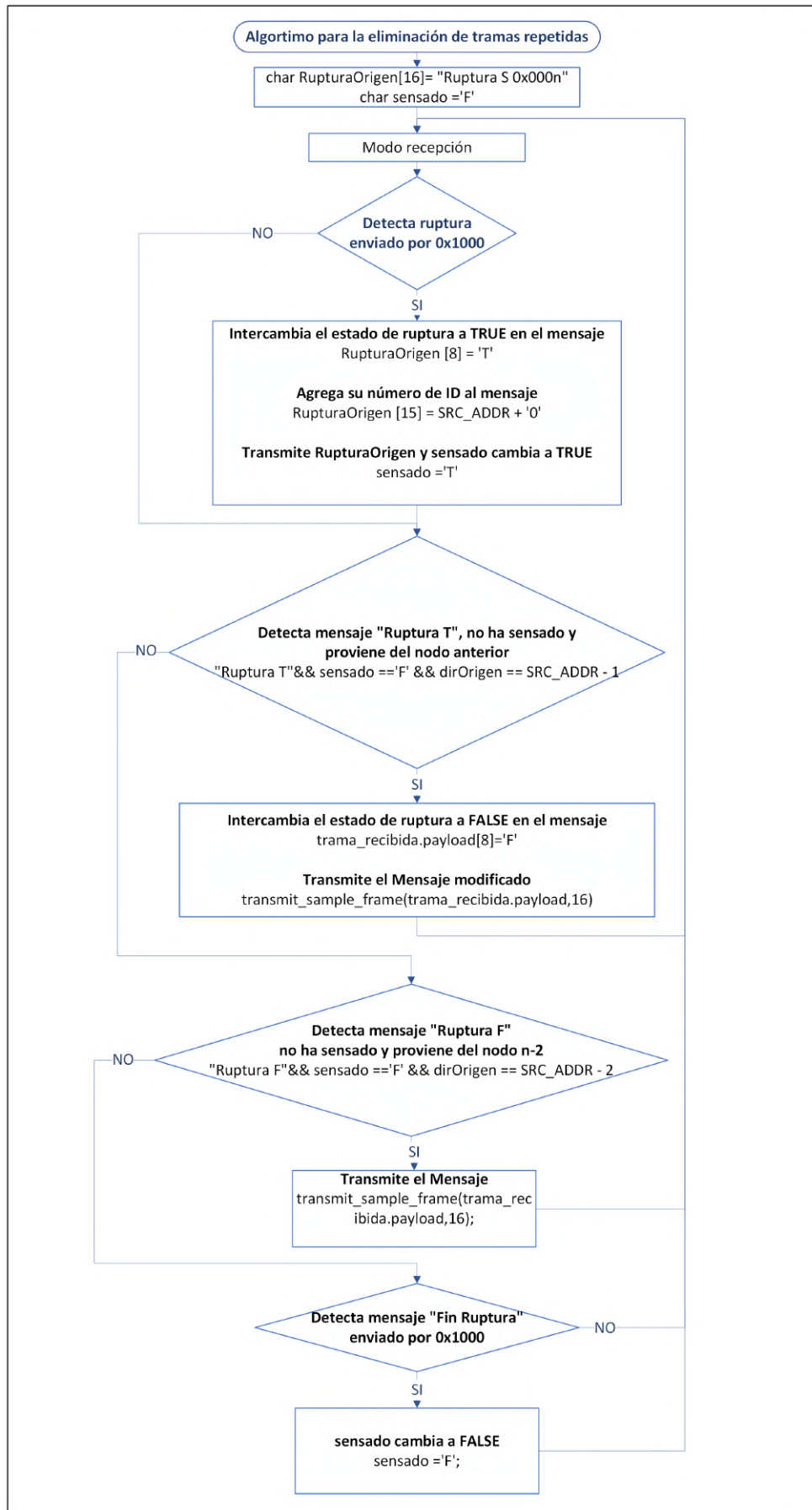


Figura 2.8: Funcionamiento del algoritmo y su Diagrama de flujo.

2.3 REQUERIMIENTOS NECESARIOS

2.3.1 Hardware necesario para la implementación

Para la instalación del algoritmo en cada uno de los nodos ATZB-256RFR2-XPRO, se necesitó de los siguientes componentes complementarios:

2.3.1.1 ATZB-256RFR2-XPRO y Alimentación

El transceptor contiene al microcontrolador Atmega256RFR2, y soporta el estándar IEEE 802.15.4, su placa se puede acoplar a un módulo extra de evaluación mediante su cabecera, pero también funciona de forma independiente con una fuente de alimentación de 3 voltios, en este caso generados por dos pilas AAA recargables [11]. Observar la Figura 2.9.

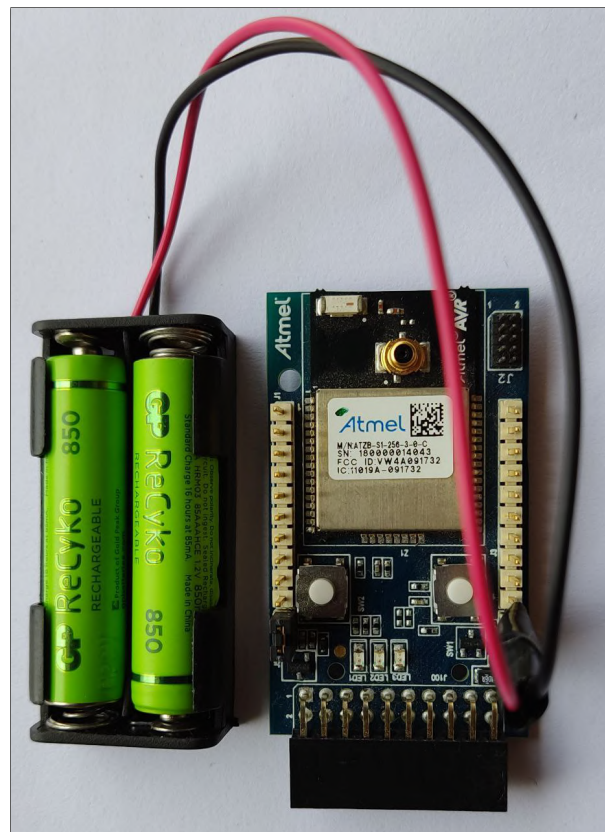


Figura 2.9: ATZB-256RFR2-XPRO y Alimentación.

2.3.1.2 Atmel-ICE Debugger

Atmel-ICE es una poderosa herramienta de desarrollo para depurar y programar microcontroladores con capacidad de depuración en chip [12]. Es compatible con:

- Programación y depuración en chip de todos los Atmel AVR de 32 bits microcontroladores en las interfaces JTAG y aWire[12].
- Programación y depuración en chip de todos los Atmel AVR XMEGA® familia dispositivos en interfaces JTAG y PDI de 2 hilos[12].
- Programación (JTAG, SPI, UPDI) y depuración de todos los Atmel AVR 8 bits microcontroladores con soporte OCD en JTAG, debugWIRE o interfaces UPDI[12].
- Programación y depuración de todos los Atmel SAM ARM Cortex-M basados microcontroladores en interfaces SWD y JTAG[12].
- Programación (TPI) de todos los Atmel tiny AVR® microcontroladores de 8 bits con soporte para esta interfaz[12].



Figura 2.10: Atmel-ICE Debugger kit básico [12].

2.3.1.3 Elemento de monitoreo para capturar tramas IEEE 802.15.4 (Sniffer)

El CC2531 es compatible con ZigBee/IEEE 802.15.4 su fabricante Texas Instrument desarrolló un núcleo MCU 8051 optimizado para la banda ISM/SRD de 2,4 GHz. Este dispositivo permite aplicaciones de grado industrial al ofrecer inmunidad al ruido de última generación, funcionamiento con una cantidad de 125 grados como máximo y baja tensión; también, el CC2531 administra un gran soporte de hardware con la finalidad de manejar paquetes y guardarlos en el búfer de datos; además de las ráfagas, cifrado, autenticación, pruebas del canal con indicación de calidad de enlace e información de temporización de paquetes [13].

El monitoreo y captura de tramas es fundamental para analizar el funcionamiento de una comunicación entre nodos, su red y comportamiento. La herramienta que se utiliza es el CC2531 USB Dongle (Sniffer), Figura 2.11, el cual captura las tramas IEEE 802.15.4 enviadas en tiempo real y pertenecientes a cualquier señal cercana, para su uso en este proyecto se decidió configurar sus parámetros de observación para captar mensajes enviados en la banda de 2.4 GHz, dentro del canal 0x0B, utilizando la herramienta Packet Sniffer.



Figura 2.11: CC2531 USB Dongle [13].

2.3.2 Software necesario para la implementación

La implementación del algoritmo en topologías lineales requiere la plataforma Atmel 7.0 que usa lenguaje C/C++ para desarrollar código de programación para algoritmos, incluidas las extensiones de Wireless Composer que usan funciones del estándar IEEE 802.15.4. También necesita la aplicación Packet Sniffer, esto le permite ver los datos generados por la tarjeta ATZB-256RFR2-XPR. Los programas Microchip Studio con el paquete Atmel 7.0 y Packet Sniffer se instalaron en el sistema operativo Windows 11.

2.4 CODIFICACIÓN DEL ALGORITMO EN EL SOFTWARE

2.4.1 Inicio del proyecto

2.4.1.1 Configuración inicial

El manual de instalación de Microchip Studio con el paquete Atmel Studio 7.0 y extensiones se agrega en el Anexo II. Al iniciar el programa en la ventana principal se encuentra el texto que dice: crear un nuevo proyecto Figura 2.12.

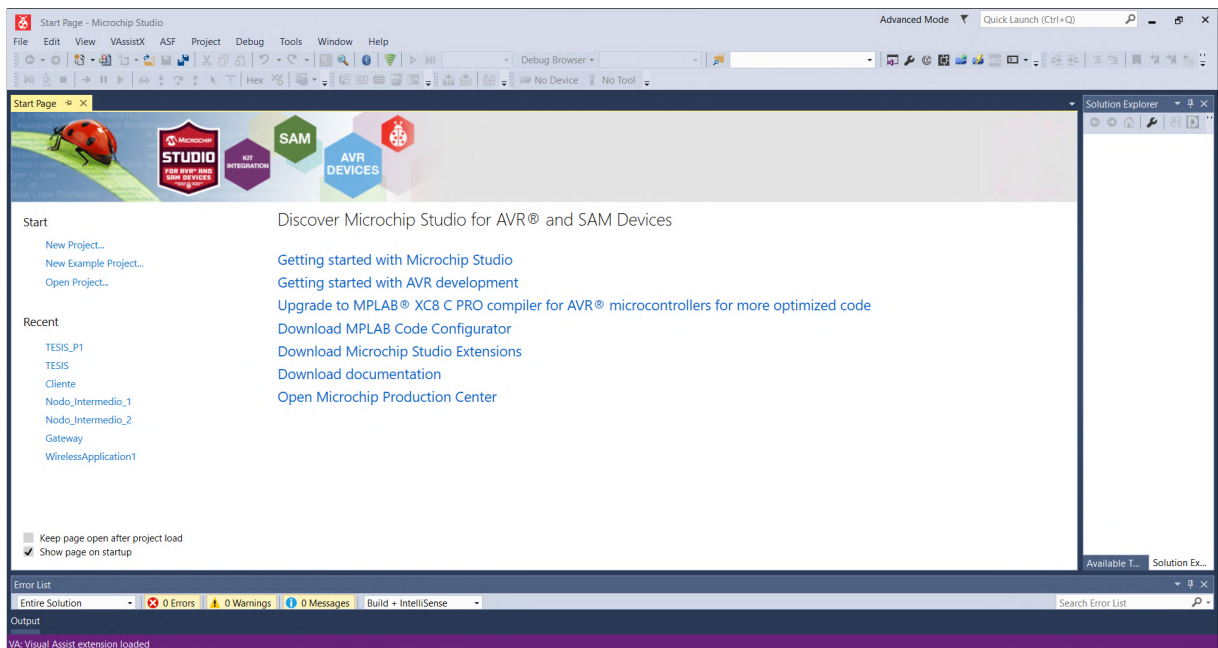


Figura 2.12: MicrochipStudio.

Después de seleccionar la opción "nuevo proyecto", se despliega la ventana que posee distintas opciones de creación Figura 2.13. Para comenzar se selecciona la opción "GCC C IEEE 802.15.4 MAC Wireless Executable Project", la cual alberga todas las características modificables del estándar IEEE 802.15.4, para finalizar con esta parte se agrega un nombre al proyecto y se lo guarda dentro del equipo.

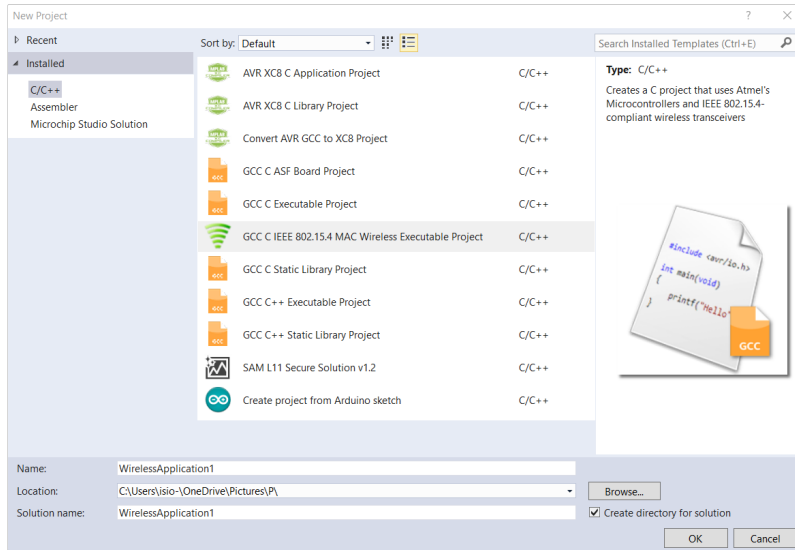


Figura 2.13: Nuevo proyecto en MicrochipStudio.

A continuación, se debe seleccionar el tipo de proyecto y la opción "Generic Project", Figura 2.14 ya que permite iniciar con el código por defecto, los dispositivos y módulos principales.

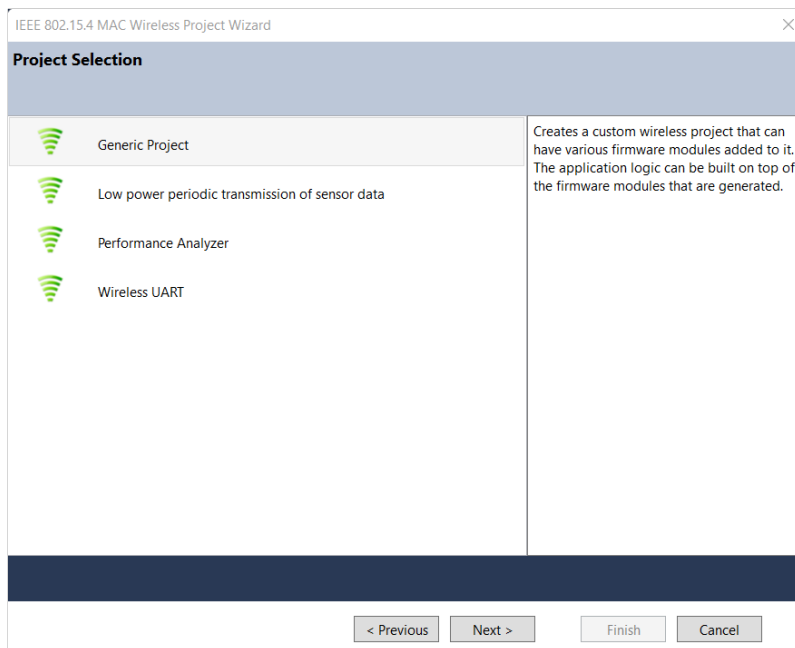


Figura 2.14: Tipo de proyecto.

Después, aparece una ventana, Figura 2.15 para la selección del microcontrolador, en este proyecto se trabaja con Atmega256RFR2, antes mencionado.

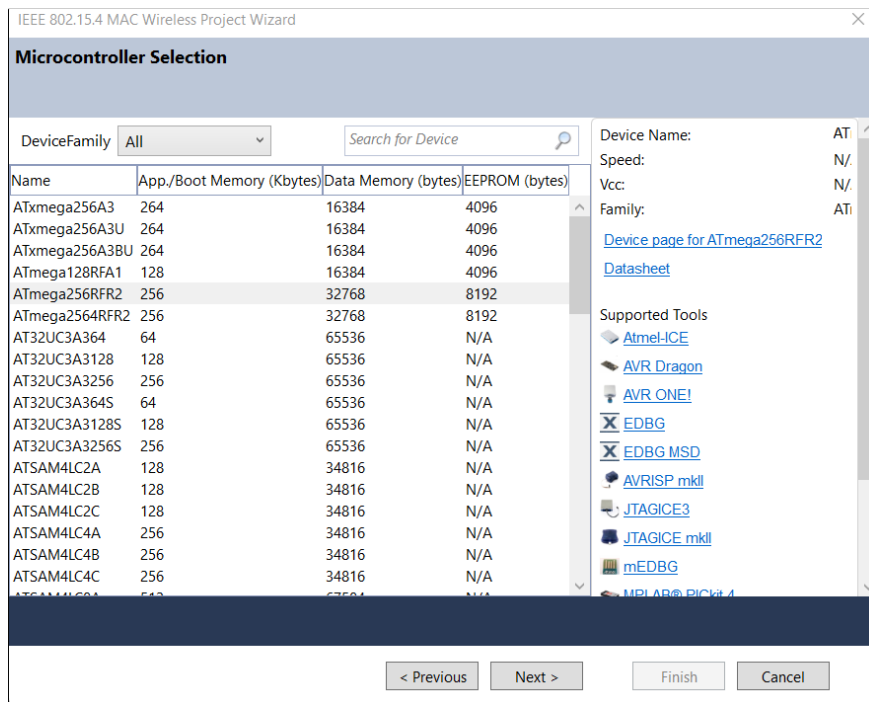


Figura 2.15: Tipo de microcontrolador.

En la siguiente ventana, Figura 2.16 es posible modificar varias configuraciones como: la velocidad de transmisión, el canal de radio frecuencia, la dirección o ID del nodo, el identificador de red PAN ID, estas últimas se pueden modificar posteriormente dentro del código..

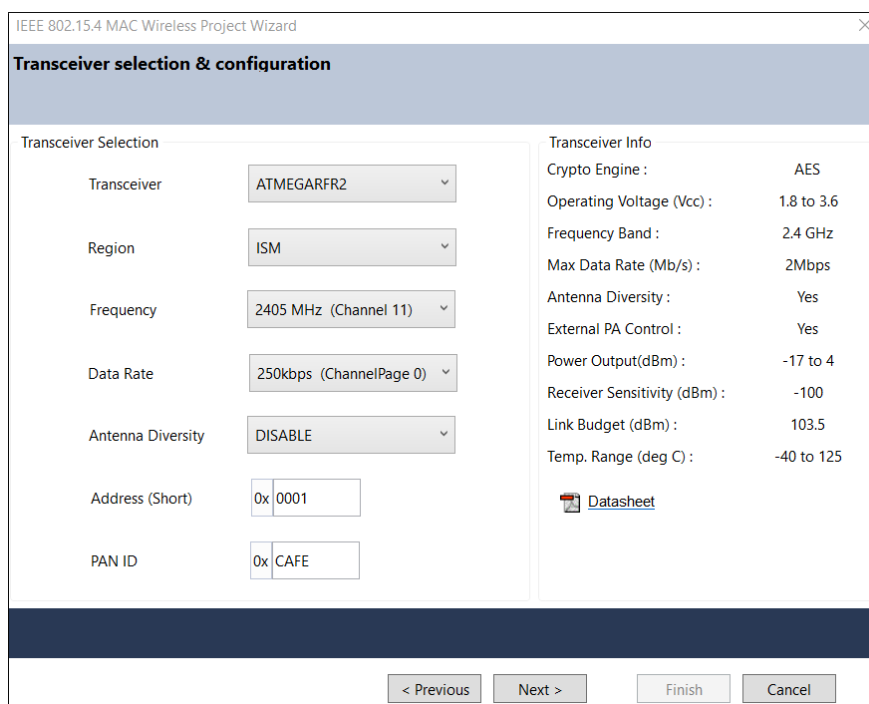


Figura 2.16: Configuración del Transceptor.

La Figura 2.17 despliega la ventana de configuración de transmisión y recepción, en este caso se selecciona la opción "Broadcast", en la cual la señal puede llegar a varios nodos y CSMA-CA es el mecanismo de gestión del medio. Finalmente, se selecciona las opciones "siguiente", y "finalizar", con lo cual ya tendremos acceso a la interfaz de edición inicial.

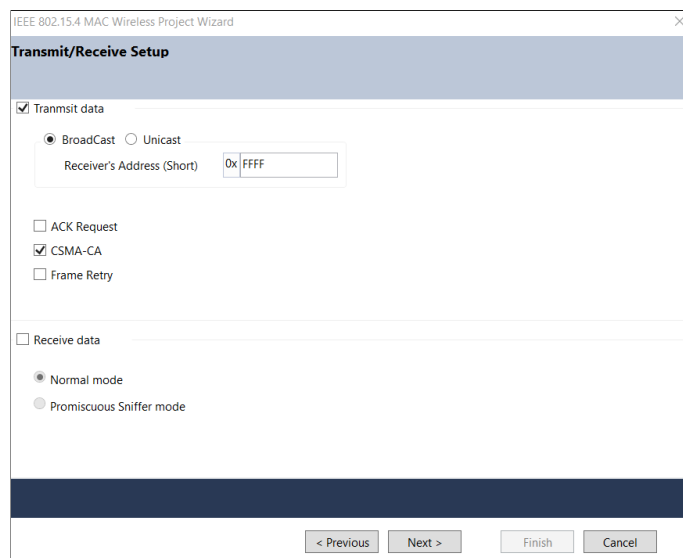


Figura 2.17: Configuración de transmisión y recepción.

La Figura 2.18 despliega la ventana de edición, la cual posee varias áreas como: explorador de soluciones (donde se encuentran los módulos y archivos), el área de Error List y Task List cada una con información de la compilación del programa.

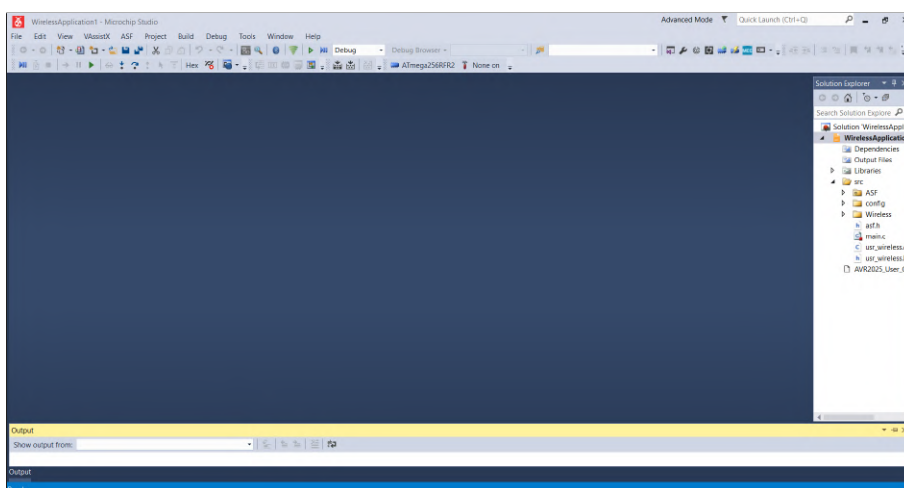


Figura 2.18: Ventana de edición.

2.4.1.2 Carga del código al nodo

Como se puede observar en la Figura 2.19 para poder cargar el programa con el algoritmo diseñado dentro del nodo, es necesario haber compilado haciendo clic en el icono o en la opción de built y built solution; después, al seleccionar la opción Tools se despliega varias opciones, entre ellas Device Programming.

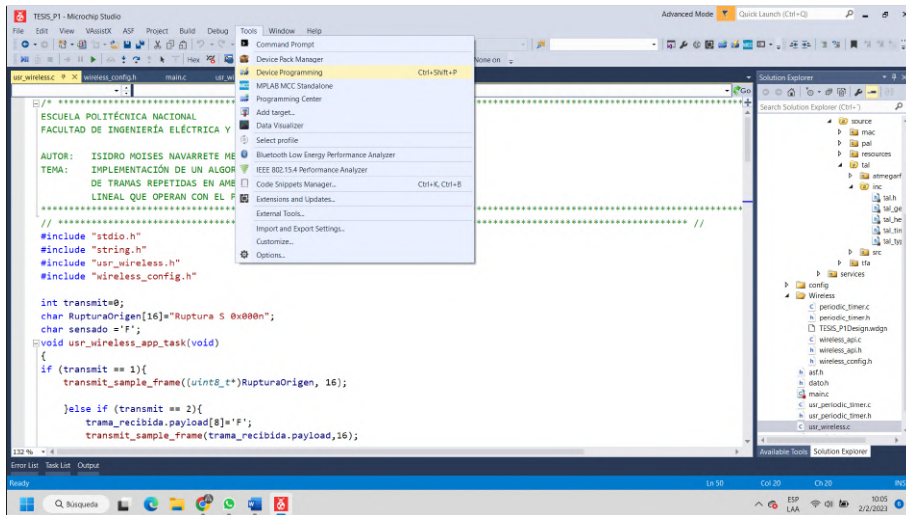


Figura 2.19: Ventana Tools.

Al hacer clic en Device Programming se mostrará Tool, Device, Interface, Device signature y Target Voltage, posteriormente se conecta Atmel-ICE Debugger y será posible seleccionarlo en la opción de Tool y Device, luego se selecciona la opción Apply lo que desplegara varias opciones Figura 2.20.

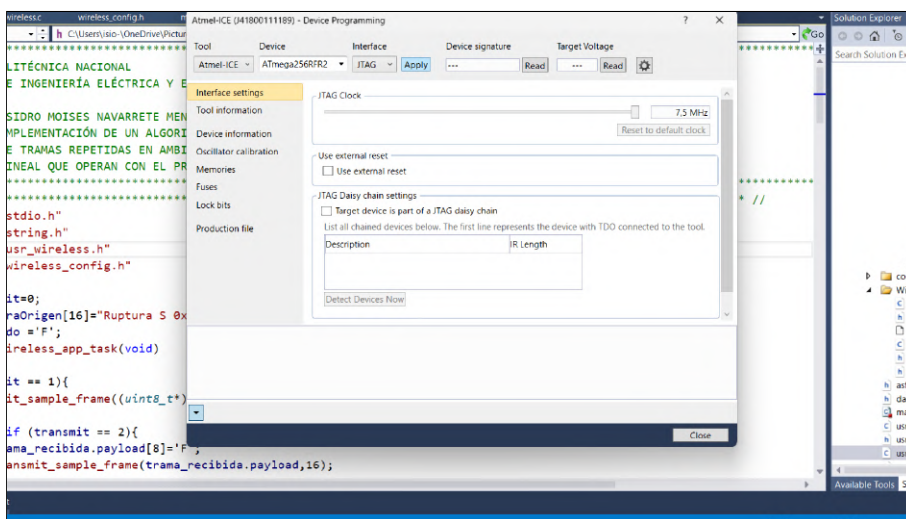


Figura 2.20: Ventana Tools.

Al conectar el nodo al Atmel-ICE Debugger, será posible seleccionar las opciones de Read, lo cual mostrara el nivel de voltaje y el nombre del nodo, Figura 2.21.

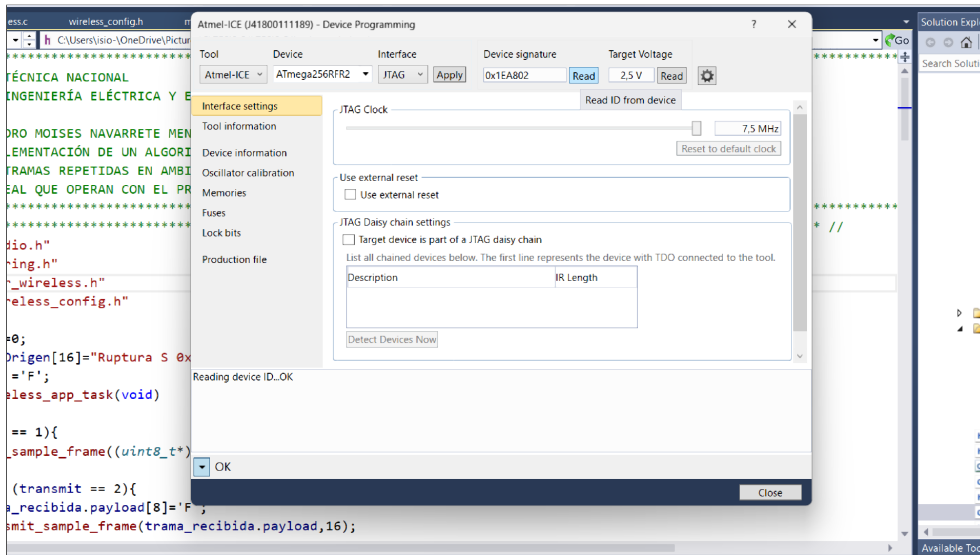


Figura 2.21: Ventana Tools.

Finalmente para cargar el código seleccionamos en el lado izquierdo la opción Memories, y después en la opción program Figura 2.22

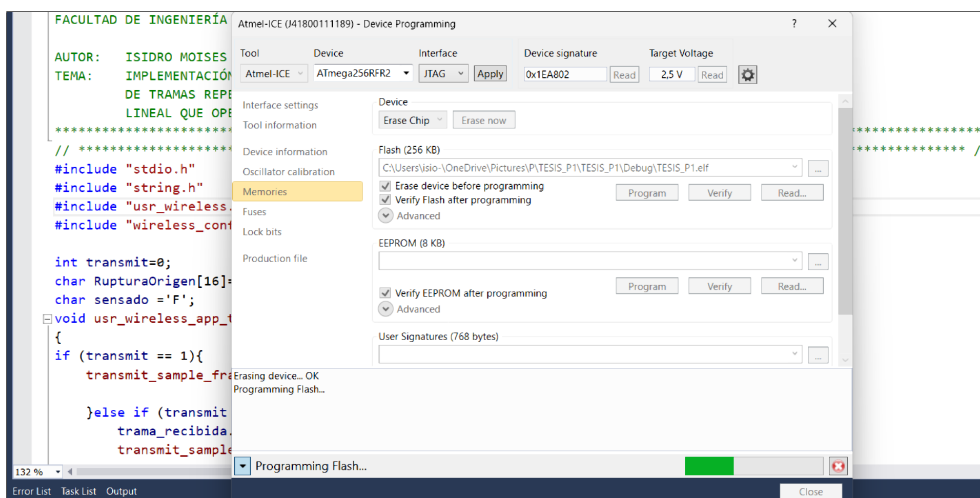


Figura 2.22: Ventana Tools.

2.4.2 Programación del algoritmo en el software

2.4.2.1 Librerías utilizadas

Las librerías utilizadas para el desarrollo del proyecto se muestran a continuación, estas dos librerías adicionales son muy conocidas "stdio.h" y "string.h", las otras dos aparecen al crear un nuevo proyecto con los nombres "usr_wireless.h" y "wireless_config.h".

```
#include "stdio.h"//Prototipos de las funciones que manipulan los datos de
    entrada y salida
#include "string.h"//Contiene un conjunto de funciones para manipular cadenas
#include "usr_wireless.h"//Permite inicializar al nodo y sus funciones de
    operacion
#include "wireless_config.h"// Da acceso a los archivos modificables que
    contienen las características de transmisi n del nodo.
```

Código 2.1: Librerías

2.4.2.2 Función principal (main)

La función (main) ubicada en el archivo main.c permite ejecutar todos los códigos del programadas por defecto en el nodo, dentro del main existe un bucle while el cual se ejecutará continuamente para que el nodo reciba y transmita.

```
/* === IMPLEMENTATION =====*/
/**Funci n principal de la aplicaci n**/
int main(void)
{ /** Inicializa el m dulo inal mbrico**/
    wireless_init();
    /*Los m dulos seleccionados en el asistente se inicializan aqu */
    modules_init();
    while (1)/* Ejecuta sus funciones continuamente*/
    {
        WirelessTask();
    }
}
```

Código 2.2: main.c

2.4.2.3 Archivo `usr_wireless.c`

`usr_wireless.c` aparece después de crear el proyecto, su función `usr_wireless_app_task`, es invocada desde la función principal `main`, esto mantiene listo al nodo para poder hacer todas las funciones de recepción y transmisión, el algoritmo diseñado y propuesto se desarrolla dentro de él, los mensajes recibidos o enviados deben seguir los procesos mencionados antes, posteriormente se presentará al algoritmo y en el Anexo 1 se encuentra el código completo.

2.4.2.4 Archivo `usr_wireless.h`

En este archivo se crea una estructura para almacenar en orden de cada uno de la información de la trama, esto se muestra a continuación.

```
#ifndef USR_WIRELESS_H_
#define USR_WIRELESS_H_

#include "wireless_api.h"

#define max_dato 101
typedef struct
{
    uint8_t longT;           // Longitud de Trama
    uint16_t fcf;           // fcf
    uint8_t num_s;          // Numero secuencia
    uint16_t dir_PAN;       // Direccion PAN
    uint16_t d_dstn;        // Direccion ORIGEN
    uint16_t d_orgn;        // Direccion DESTINO
    uint8_t Payload_IEE_802_15_42[max_dato]; // Payload
    uint16_t fcs;           // frame control sequence
} trama_ieee_802_15_42;

trama_ieee_802_15_42 trama_recibida;
```

Código 2.3: `usr_wireless.h`

2.4.2.5 wireless_api.c y wireless_api.h

wireless_api.c y wireless_api.h se encuentran dentro de la carpeta Wireless, su función es contener la reunión de listas de archivos fuente provenientes de cada módulo del proyecto; también, se generan funciones básicas explicadas a continuación:

- **wireless_api.c:** En este archivo se puede configurar la potencia de transmisión; además, se encuentra la función de transmisión de tramas, mostrada a continuación y en la cual se añade el parámetro `uint16_t`, con el fin de enviar tramas a una dirección en especial, esto es útil si se utiliza transmisión unicast.

```
void transmit_sample_frame(uint8_t* payload, uint8_t payload_length)
{
    static uint16_t seq_num = 0;
    bool ack_req = ACK_REQ;
    bool frame_retry = FRAME_RETRY;
    csma_mode_t csma_mode = CSMA_MODE;
    uint16_t dst_addr = CCPU_ENDIAN_TO_LE16((uint16_t)DST_ADDR);

    transmit_frame(
        FCF_SHORT_ADDR,
        (uint8_t*)&dst_addr,
        FCF_SHORT_ADDR,
        seq_num++,
        payload,
        payload_length,
        ack_req,
        csma_mode,
        frame_retry);
}
```

Código 2.4: wireless_api.c

- **wireless_api.h:** En el se puede agregar parámetros de configuración inicial de la transmisión de trama, no es necesario modificar este archivo.

2.4.2.6 Nodo externo

En el siguiente código se implementó el algoritmo diseñado y explicado con anterioridad, este código se encuentra dentro del proyecto "Nodo Externo", que a su vez contiene el archivo `usr_wireless.c`.

El algoritmo contiene una variable de conteo que se modifica dependiendo del mensaje enviado, además de un tiempo de espera de 5 segundos entre mensajes, todo esto se ubica dentro de la "función `usr_wireless_app_task`", mostrada a continuación.

```
#include "stdio.h"
#include "string.h"
#include "usr_wireless.h"
#include "wireless_config.h"

int contador=0; //Variable que ayuda a enviar dos mensajes diferentes
uint8_t payloadR[7]; //Longitud del mensaje Ruptura
/**Esta función deber a editarse por el usuario para agregar tareas de
aplicación*/
void usr_wireless_app_task(void)
{
    // (Asistente de proyectos): agregue tareas de aplicación aquí .
    // Esta función será llamada repetidamente desde main.c. (Consulte la
    función app_task(), WirelessTask() en main.c)

    if (contador==0){          transmit_sample_frame((uint8_t*)"Ruptura", 7);
        contador = 1;
    }
    else if(contador==1){
        transmit_sample_frame((uint8_t*)"Fin Ruptura", 11);
        contador = 0;
    }
    delay_ms(5000);
}

void usr_frame_received_cb(frame_info_t *frame)
{}

void usr_frame_transmitted_cb(retval_t status, frame_info_t *frame)
{}
```

Código 2.5: Algoritmo nodo externo

2.4.2.7 Algoritmo

En el siguiente código se implementó el algoritmo para la eliminación de tramas repetidas, este fue diseñado y explicado con anterioridad, a continuación, se encuentra el proyecto "Algoritmo para la eliminación de tramas repetidas", que a su vez contiene el archivo `usr_wireless.c`.

El algoritmo contiene dos variables char que se modifican dependiendo del caso a analizar, todo esto se ubica dentro de la "funcion void `usr_frame_received_cb`", mostrada a continuación

```
#include "stdio.h"
#include "string.h"
#include "usr_wireless.h"
#include "wireless_config.h"
int tiempo=1;
char RupturaOrigen[16]="Ruptura S 0x000n";
char sensado ='F';
void usr_wireless_app_task(void)
{}
// Funcion de recepcion //
void usr_frame_received_cb(frame_info_t *frame)
{
    memset(&trama_recibida,0,sizeof(trama_recibida)); // Reserva el espacio de
    memoria
    memcpy(&trama_recibida,frame->mpdu,sizeof(trama_recibida)); // Copia la
    informacion de la memoria fuente(buffer) al destino
    bmm_buffer_free(frame->buffer_header); //Elimina los datos del buffer, evita
    superpocicion.

    uint16_t dirOrigen; //Direccion del mensaje Recibido
    uint8_t msgTypeMQTT_SN; //Tipo de mensaje MQTT-SN
    uint8_t longitudMQTT_SN; //Longitud del mensaje MQTT-SN
    dirOrigen=trama_recibida.d_orgn;
    longitudMQTT_SN= trama_recibida.payload[0];
    msgTypeMQTT_SN= trama_recibida.payload[1];
```

Código 2.6: Algoritmo parte 1

```

if (dirOrigen==0x1000 && strcmp((const char *) trama_recibida.payload, "
    Ruptura", 7) == 0){ // Recibe la direccion origen del nodo
    // ## Mediante la variable msgTypeMQTT_SN se procede a comprobar el tipo
    de mensaje recido.
    RupturaOrigen [8] = 'T'; // variable que determina si este nodo recibio la
    alerta
    RupturaOrigen [15] = SRC_ADDR + '0'; //agrega su direccion al mensaje
    transmit_sample_frame((uint8_t*)RupturaOrigen, 16); //transmite el mensaje
    sentido = 'T'; //variable que determina si este nodo senso
    bmm_buffer_free(frame->buffer_header); //Elimina los datos del buffer, evita
    superpocicion.
    //en el caso de que reciba mensaje de ruptura de un nodo a su alcance, se
    realiza las siguientes comprobaciones:

    //1.-si llega de un nodo que ya senso, no he sentido y es el anterior nodo,
    transmito

}else if(strcmp((const char *) trama_recibida.payload, "Ruptura T", 9)==0
    && sentido == 'F' && dirOrigen == SRC_ADDR - 1){
    trama_recibida.payload[8]='F'; //Modifico el mensaje para decir que no he
    sentido
    transmit_sample_frame(trama_recibida.payload,16);
}else if(strcmp((const char *) trama_recibida.payload, "Ruptura F", 9)==0
    && sentido == 'F' && dirOrigen == SRC_ADDR - 2){
    //si llega de un nodo que no senso, no he transmitido, y es el nodo
    anterior anterior, transmito y transmitir es TRUE
    transmit_sample_frame(trama_recibida.payload,16);
}else if (dirOrigen==0x1000 && strcmp((const char *) trama_recibida.
    payload, "Fin Ruptura", 11) == 0){

    sentido = 'F';
    bmm_buffer_free(frame->buffer_header);}
}

void usr_frame_transmitted_cb(retval_t status, frame_info_t *frame)
{}
}

```

Código 2.7: Algoritmo parte 2

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 CONFIGURACIONES INICIALES

El desarrollo del algoritmo diseñado para la eliminación de tramas repetidas dentro de una topología lineal está implementado en todos los nodos pertenecientes al prototipo propuesto anteriormente, seguidamente se ejecutaron las pruebas pertinentes para comprobar su correcto funcionamiento y su tiempo de respuesta para cada uno de los nodos; además, se generó el escenario en el que se puede ver el efecto de no aplicar el algoritmo en los nodos. El código completo se encuentra adjunto en el Anexo I.

3.1.1 Elementos usados en las pruebas

Los Elementos necesarios para ejecutar todas y cada una de las siguientes pruebas, se observan en la Tabla 3.1.

Tabla 3.1: Elementos del prototipo implementado

Elementos	Cantidad
Transceptor TZB-256RFR2-XPRO	7
Computador	1
Sniffer CC2531 USB	2
Atmel-ICE Debugger	1
Cable extensor USB	1

El diagrama y la topología lineal del prototipo implementado se encuentra en la Figura 3.1.

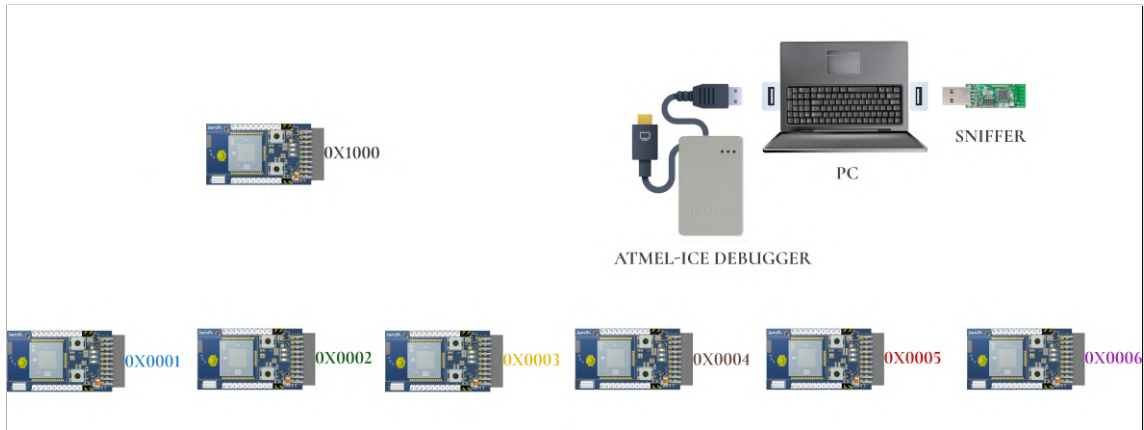


Figura 3.1: Diagrama del prototipo implementado.

3.1.2 Parametros iniciales en el software

Con el proposito de capturar las tramas enviadas por los nodos o sensores, es necesaria la herramienta de monitoreo Sniffer y su software Packet Sniffer, al cual es importante modificar el canal que se va a observar y el tipo de estandar a utilizar Figura 3.2 y Figura 3.3.

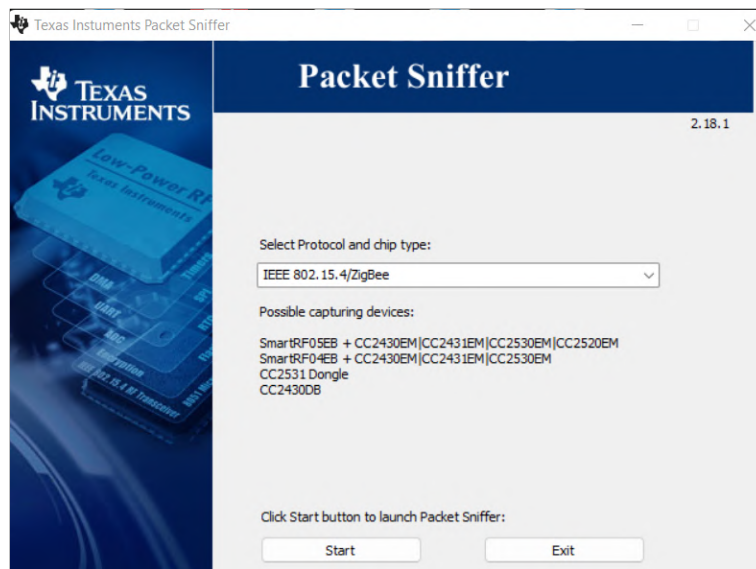


Figura 3.2: Packet Sniffer primera configuración.

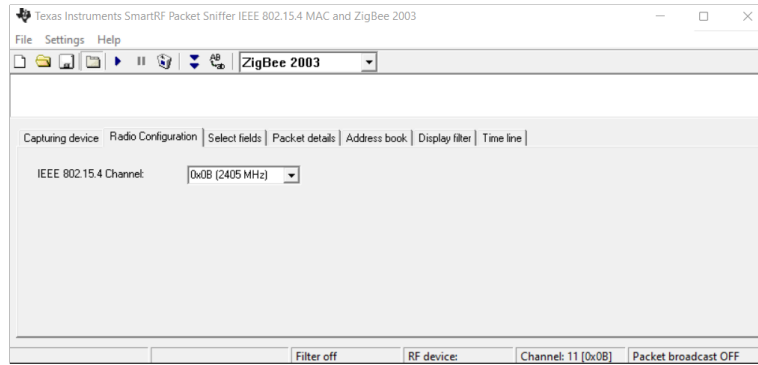


Figura 3.3: Packet Sniffer primera configuración.

3.1.3 Topología Lineal

Las pruebas se realizaron en un ambiente controlado, es decir en una mesa normal debido a que el algoritmo no depende de la distancia Figura 3.4.

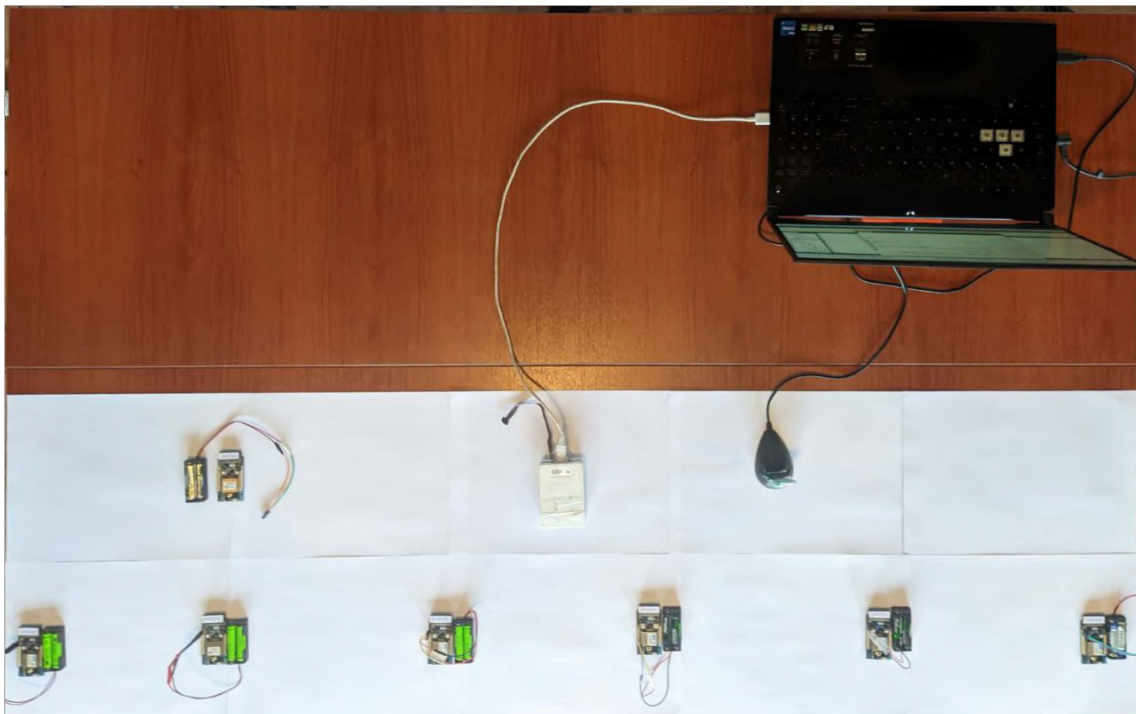


Figura 3.4: Prototipo Implementado.

El Sniffer y el programador se encuentran conectados a la computadora, permitiendo modificar el código de cada nodo de forma más rápida, cabe mencionar que cada nodo debe ser cargado con el mismo algoritmo pero con su ID respectiva Figura 3.5 y Figura 3.6.

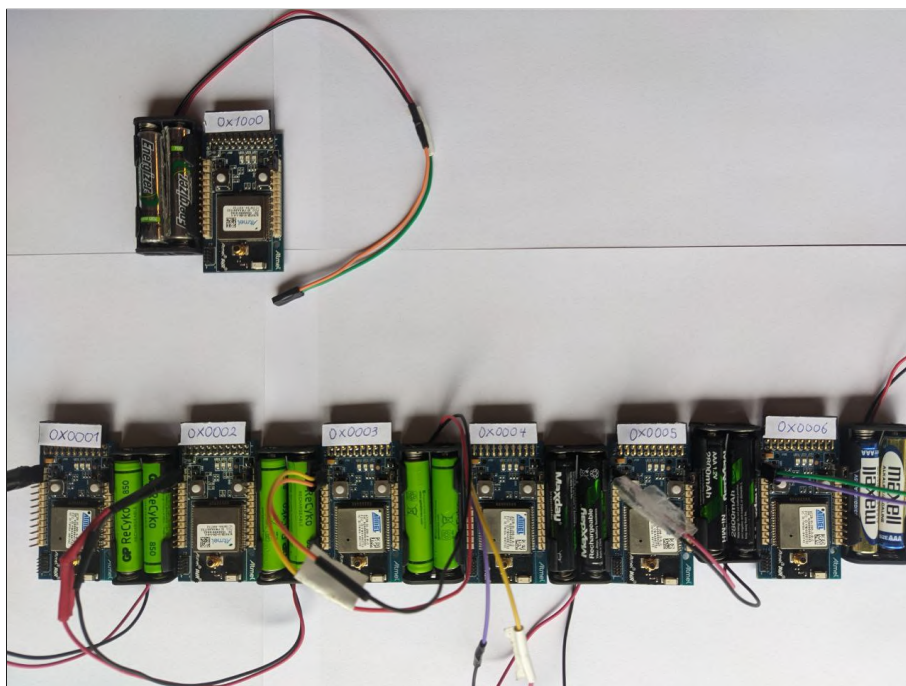


Figura 3.5: Secuencia de nodos.



Figura 3.6: Conexión a computadora.

3.2 PRUEBAS EJECUTADAS

3.2.1 Prueba 1: Transmisión realizada al mismo tiempo

En esta prueba se utilizó el nodo externo para enviar el mensaje de ruptura, este mensaje puede ser desencadenado por cualquier otra anomalía que los nodos puedan detectar, por ejemplo, con el uso de sensores. El mensaje de ruptura será detectado por los tres primeros nodos, los cuales retransmitirán la ruptura casi al mismo tiempo; además, para observar las tramas enviadas se utiliza dos Sniffers idénticos ya que es necesario comprobar que todos los nodos envíen correctamente los mensajes Figura 3.7. el nodo externo transmite dos mensajes de ruptura con 5 segundos de separación. En el primer mensaje, ambos sniffers capturaron las mismas tramas en tiempos muy similares, pero en el segundo mensaje como se muestra en la Figura 3.8 los mensajes enviados por los nodos 0X0003 Y 0X0001 no son capturados adecuadamente y están representados con una línea de puntos, la primera trama no es capturada por el primer sniffer y la segunda trama no es capturada por el segundo sniffer, al observar los tiempos en los cuales las tramas fueron capturadas, mostrados en la Tabla 3.2 se puede notar una muy pequeña diferencia, es decir que los sniffers no pudieron capturar una de las dos tramas, esto comprueba que las tramas si fueron enviadas aunque no hayan sido capturadas por el mismo sniffer, la causa más probable es la sensibilidad del equipo al momento de intentar capturar dos tramas enviadas casi al mismo tiempo.

Tabla 3.2: Tiempos de captura prueba 1.

Nodos	Sniffer 1 Tiempo [ms]	Sniffer 2 Tiempo [ms]	Tiempo promedio [ms]
0x1000	0	0	0
0x0001	1.393	-	T1 = 1.393
0x0003	-	1.381	T3 = 1.381
0x0002	4.606	4.615	T2 = 4610.5

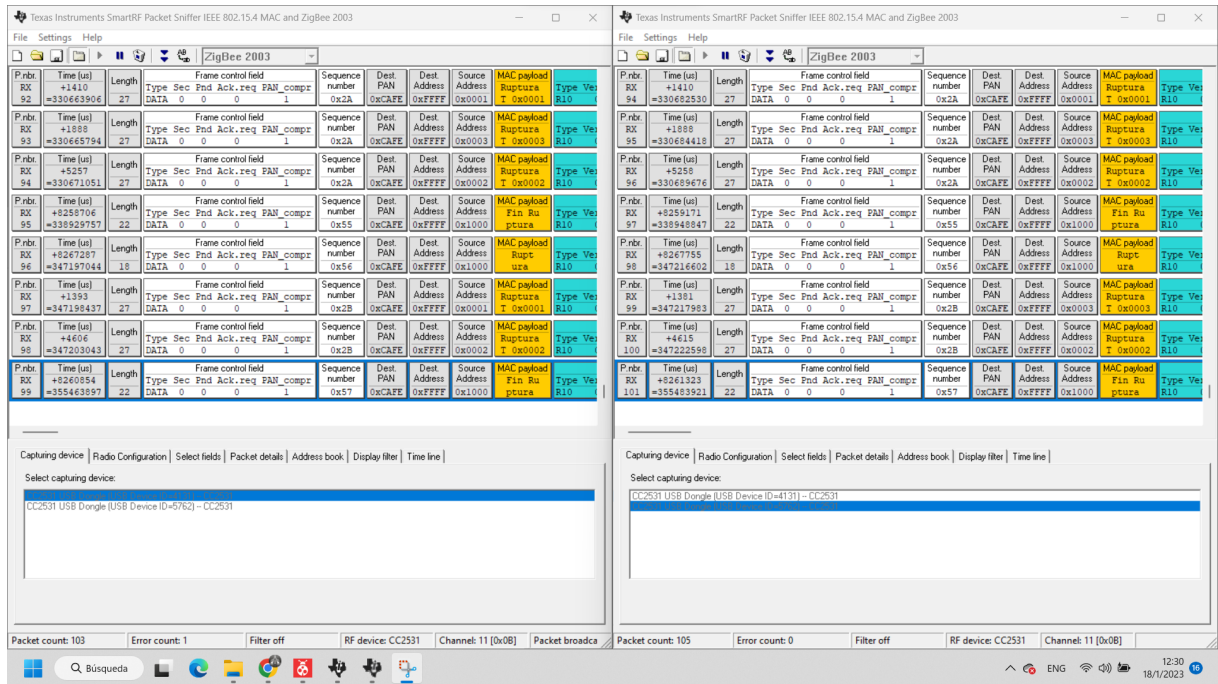


Figura 3.7: Capturas de mensajes prueba 1 Sniffers.

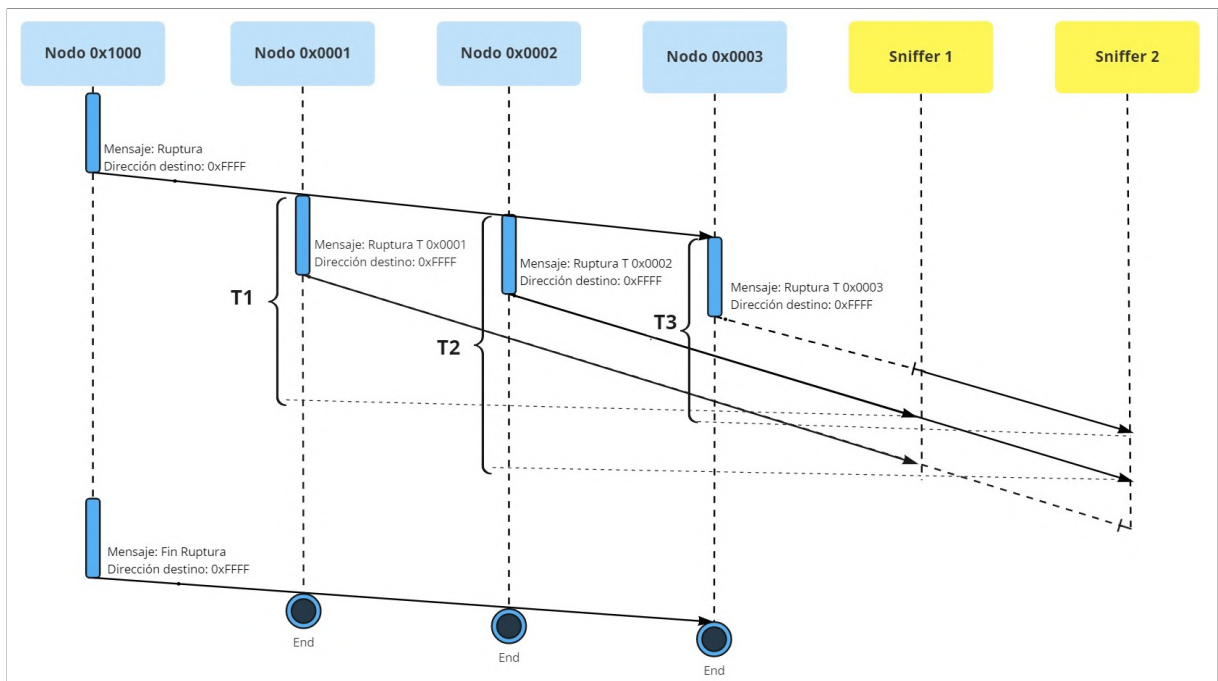


Figura 3.8: Transmisión de nodos prueba 1.

3.2.2 Prueba 2: Transmisión sin el uso del algoritmo diseñado

Una vez que los nodos transceptores están ordenados, se cargó el código donde se eliminó la parte del algoritmo diseñado, esto se realizó para evidenciar la repetición de tramas lo que genera redundancia y consume recursos. A continuación, en la Figura 3.9 y Figura 3.10, se observa la prueba realizada, y los múltiples mensajes enviados por todos los nodos, el orden en el cual se enviaron las tramas puede variar debido a que es aleatorio, esto ocurre cuando el canal no se encuentra libre y los nodos deben esperar para poder enviar su trama; además, varios nodos envían repetidamente el mismo mensaje y como se observó en la prueba anterior, también existe la posibilidad de que una trama no sea capturada pero aun así haya sido retransmitida por otro nodo, el tiempo que tomo la transmisión de los 6 nodos es de 47,576 ms.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	NWK Frame control field	NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq. num	LQI	FCS
P.RX RX 53	+8255545 =82571394	18	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x1000	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x0C	139	OK
P.RX RX 54	+1668 =82573062	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0002	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	166	OK
P.RX RX 55	+1419 =82574481	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x0003	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	169	OK
P.RX RX 56	+2143 =82576624	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x08	0xC9FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	162	OK
P.RX RX 57	+2286 =82578910	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x08	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	193	OK
P.RX RX 58	+5003 =82583913	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xC9FE	0xFFFF	0x0001	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	153	OK
P.RX RX 59	+2851 =82586764	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x08	0xC9FE	0xFFFF	0x0005	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	220	OK
P.RX RX 60	+1667 =82588431	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x09	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	193	OK
P.RX RX 61	+3981 =82592412	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0B	0xC9FE	0xFFFF	0x0003	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	169	OK
P.RX RX 62	+2596 =82595008	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x09	0xC9FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	162	OK
P.RX RX 63	+2425 =82597433	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x09	0xC9FE	0xFFFF	0x0005	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	220	OK
P.RX RX 64	+3522 =82600955	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	193	OK
P.RX RX 65	+9780 =82610735	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x07	0xC9FE	0xFFFF	0x0002	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	166	OK
P.RX RX 66	+1973 =82612708	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	165	OK
P.RX RX 67	+2608 =82615316	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x0005	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	220	OK
P.RX RX 68	+3654 =82618970	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0B	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	193	OK
P.RX RX 69	+8210797 =90829767	22	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0B	0xC9FE	0xFFFF	0x1000	Fin Ruptura	Type Version DR GA Sec R10 0x1 1 1 0 0 0 0	0x206E	0x7552	0x70	0x74	139	OK

Figura 3.9: Captura de mensajes prueba 2.

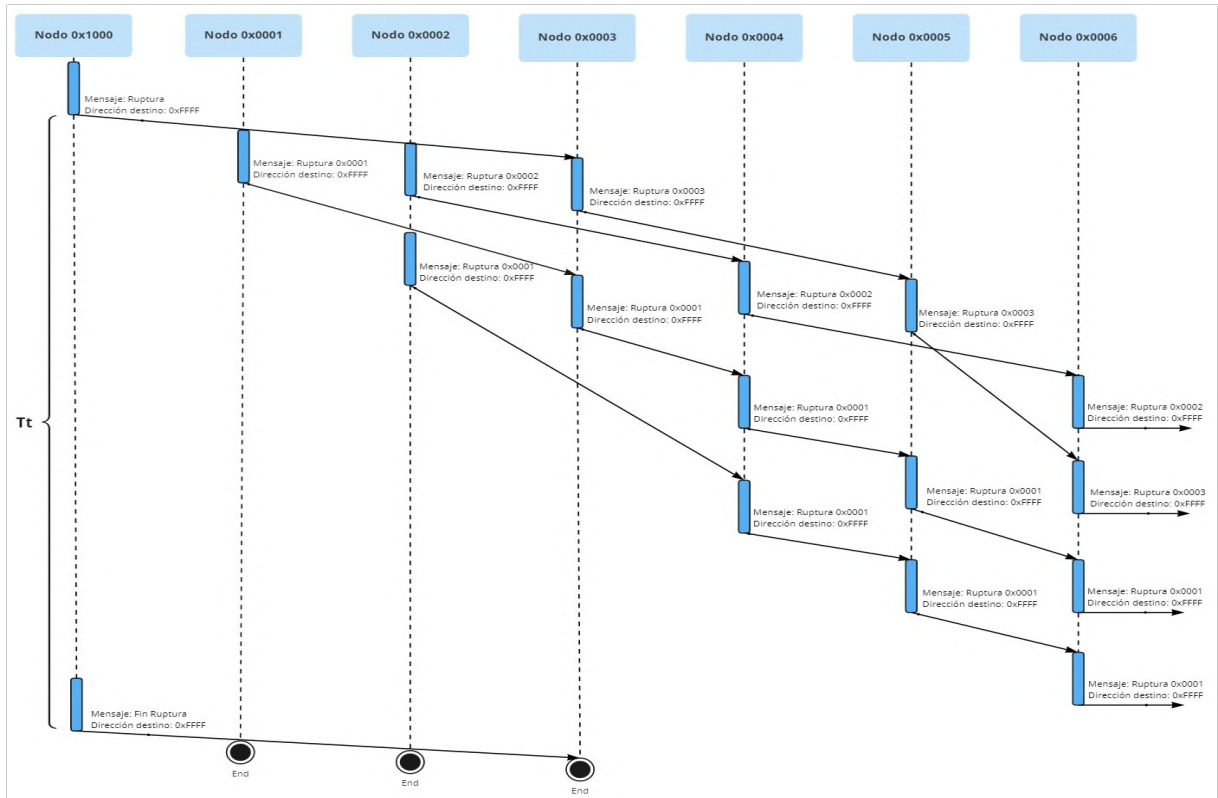


Figura 3.10: Transmision de nodos prueba 2.

3.2.3 Prueba 3: Transmisión con el uso del algoritmo diseñado y detección de ruptura por un nodo

Al cargar el código con el algoritmo diseñado, se puede evidenciar la eliminación de la repetición de las tramas, es decir de la redundancia, generando una transmisión óptima, corrigiendo los problemas mostrados en la prueba anterior; a continuación, en la Figura 3.11 y Figura 3.12, se observa la prueba realizada, y los mensajes enviados por todos los nodos; además, al contabilizar los tiempos de cada transmisión se obtiene el valor de 9,930 ms reduciendo en gran medida el tiempo de la prueba anterior.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN Address	Dest. Address	Source Address	MAC payload	NWK Frame control field	NWK Dest Address	NWK Src Address	Broadcast Radius	Broadcast Seq num	LQI	FCS
RX 1	+0	18	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xC9FE	0xFFFF	0x1000	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x1B	179	OK
RX 2	+2325	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0C	0xC9FE	0xFFFF	0x0001	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	187	OK
RX 3	+4308	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0C	0xC9FE	0xFFFF	0x0002	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	179	OK
RX 4	+1690	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0C	0xC9FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	158	OK
RX 5	+3832	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0C	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	187	OK
RX 6	+8266371	22	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xC9FE	0xFFFF	0x1000	Fin Ruptura	Type Version DR GA Sec R10 0x1 1 1 0 0 0 0	0x206E	0x7552	0x70	0x74	182	OK

Figura 3.11: Captura de mensajes prueba 3.

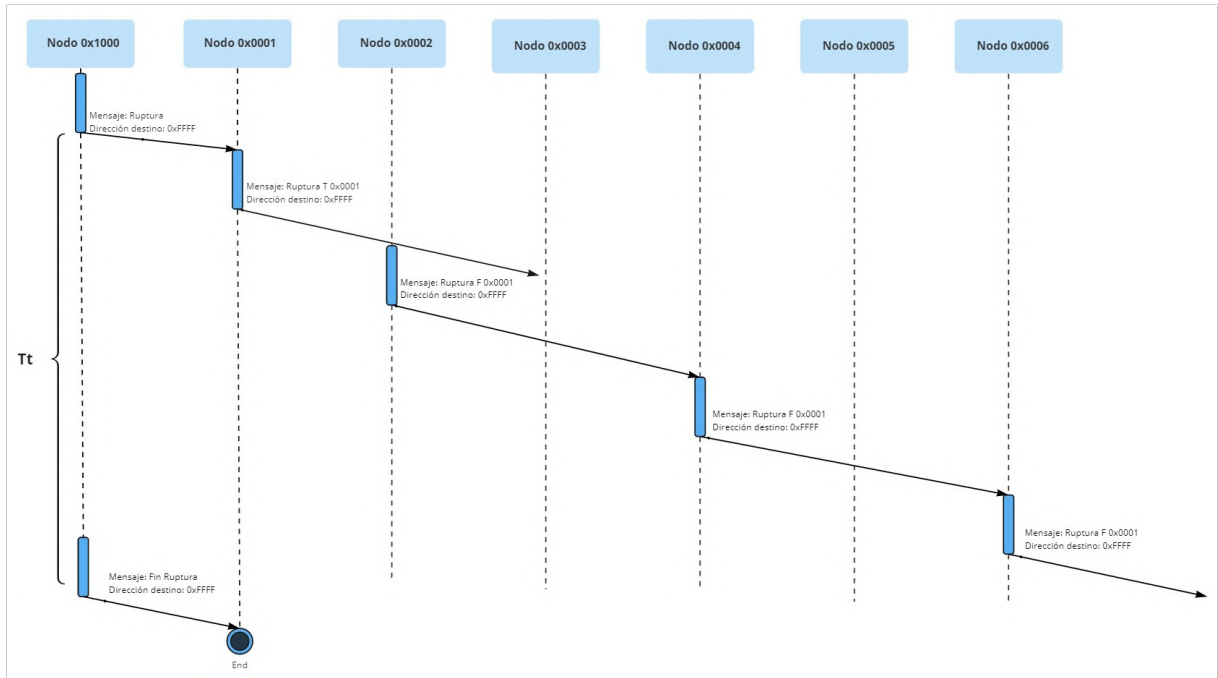


Figura 3.12: Transmisión de nodos prueba 3.

3.2.4 Prueba 4: Transmisión con el uso del algoritmo diseñado y detección de ruptura por dos nodos

En la Figura 3.13 y Figura 3.14, se observa la prueba realizada, y los mensajes enviados por todos los nodos involucrados, dando como resultado la retransmisión de un solo mensaje por parte del quinto nodo, este mensaje podrá continuar a través de los siguientes nodos, en el caso de que existan o su vez al nodo final o Gateway, el tiempo de transmisión de los nodos es de 14,599 ms.

Texas Instruments SmartRF Packet Sniffer IEEE 802.15.4 MAC and ZigBee 2003

P.nbr	Time (us)	Length	Frame control field	Sequence number	Dest. PAN Address	Dest. Address	Source Address	MAC payload	NWK Frame control field	NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq. num	LQI	FCS
RX 1	+0	18	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xC9FE	0xFFFF	0x1000	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x1B	179	OK
RX 2	+2332	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x22	0xC9FE	0xFFFF	0x0001	T 0x0001	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	182	OK
RX 3	+6667	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0A	0xC9FE	0xFFFF	0x0002	Ruptura F 0x0002	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	142	OK
RX 4	+11313	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0D	0xC9FE	0xFFFF	0x0003	Ruptura F 0x0002	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	155	OK
RX 5	+3286	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x0D	0xC9FE	0xFFFF	0x0005	Ruptura F 0x0002	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	209	OK
RX 6	+8270956	22	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xC9FE	0xFFFF	0x1000	Fin Ruptura	Type Version DR GA Sec R10 0x1 1 1 0 0 0 0	0x206E	0x7552	0x70	0x74	179	OK

Figura 3.13: Captura de mensajes prueba 4.

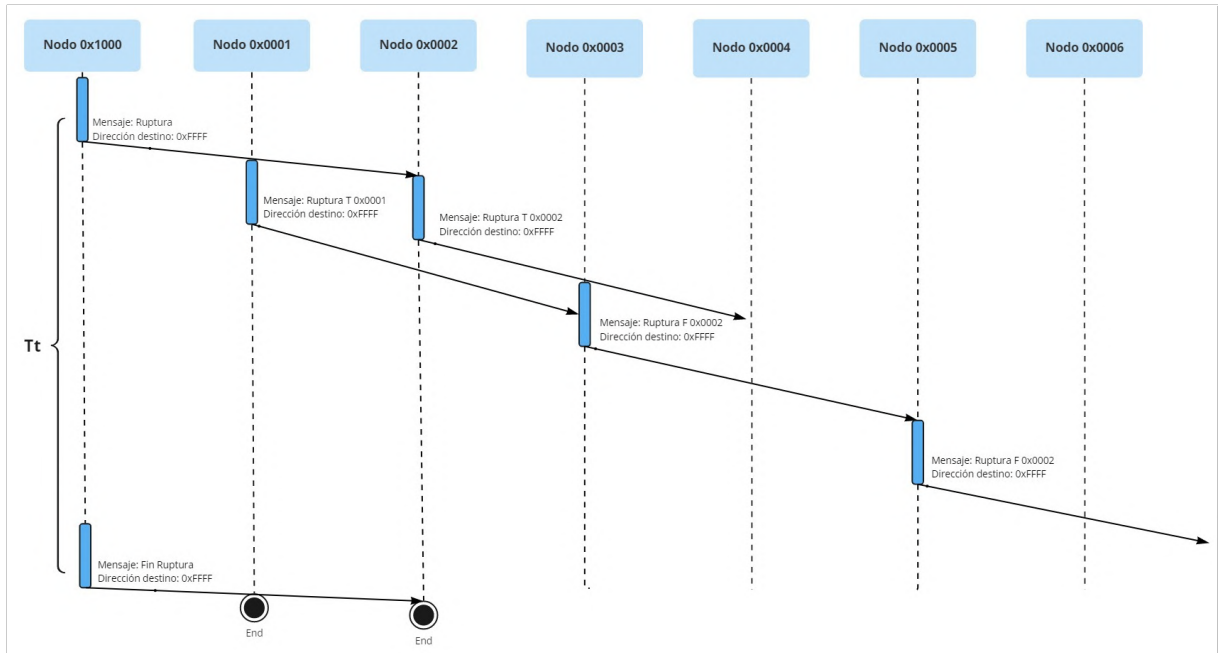


Figura 3.14: Transmisión de nodos prueba 4.

3.2.5 Prueba 5: Transmisión con el uso del algoritmo diseñado y detección de ruptura por tres nodos

La prueba realizada para tres nodos se muestra en la Figura 3.15 y Figura 3.16, estos nodos transmiten casi al mismo tiempo y gracias al algoritmo solamente se retransmite el mensaje del último nodo que detecto la ruptura, permitiendo que el sexto nodo retransmita un único mensaje y evitando por completo el envío de mensajes innecesarios, el tiempo de transmisión de los nodos es de 20,301 ms.

Texas Instruments SmartRF Packet Sniffer IEEE 802.15.4 MAC and ZigBee 2003

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	NWK Frame control field	NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq. num	LQI	FCS
P.nbr. RX 1	+0	18	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x4A	0xC4FE	0xFFFF	0x1000	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x22	198	OK
P.nbr. RX 2	+2945 =2945	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xC4FE	0xFFFF	0x0001	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	177	OK
P.nbr. RX 3	+1423 =4368	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xC4FE	0xFFFF	0x0002	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	177	OK
P.nbr. RX 4	+9080 =13448	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xC4FE	0xFFFF	0x0003	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	171	OK
P.nbr. RX 5	+3267 =16715	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x17	0xC4FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	155	OK
P.nbr. RX 6	+3586 =20301	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC4FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	161	OK
P.nbr. RX 7	+253664 =273965	22	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x4B	0xC4FE	0xFFFF	0x1000	Fin Ruptura	Type Version DR GA Sec R10 0x1 1 1 0 0 0 0	0x206E	0x7552	0x70	0x74	198	OK

Figura 3.15: Captura de mensajes prueba 5.

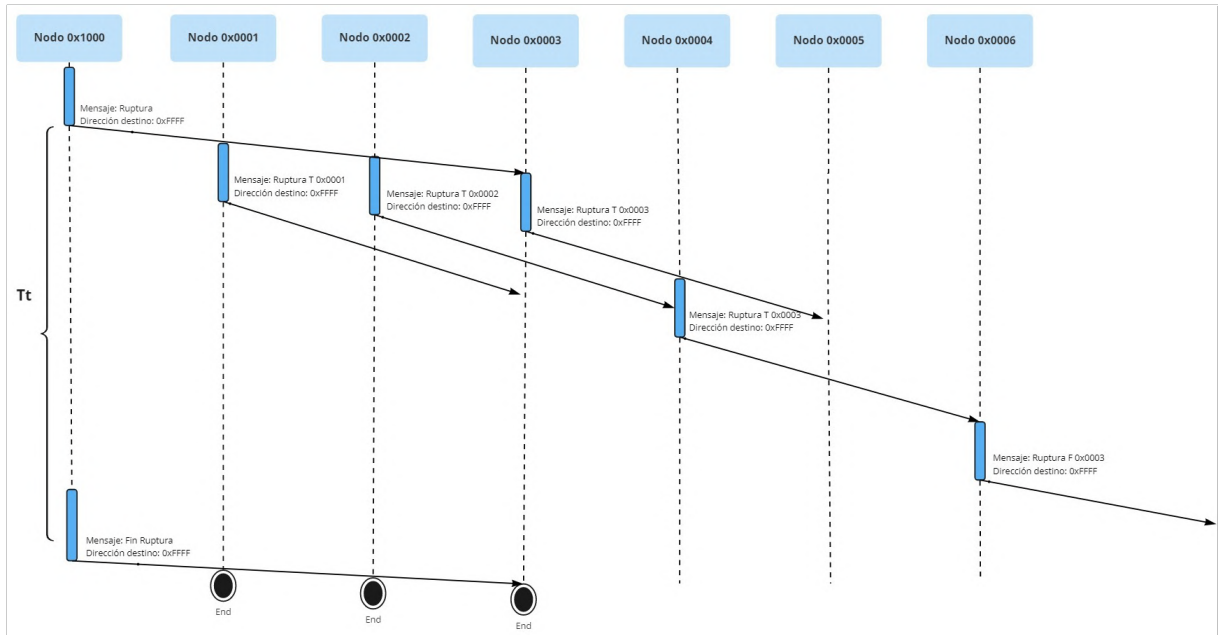


Figura 3.16: Transmisión de nodos prueba 5.

3.2.6 Prueba 6: Transmisión con el uso del algoritmo diseñado y detección de ruptura por cuatro nodos

En esta prueba se toma en consideración un caso en el que cuatro nodos transmiten casi simultáneamente, esto podría ocurrir en un evento de ruptura o anomalía a gran escala, como podría ser el caso del aumento de temperatura, presión o humedad dentro de una gran zona cubierta por estos nodos, en la Figura 3.17 y Figura 3.18, se observa las transmisiones realizadas por todos los nodos involucrados, demostrando que el algoritmo funciona correctamente a pesar de que los nodos transmitieron en desorden y casi al mismo tiempo, permitiendo que solo un mensaje sea retransmitido y obteniendo un tiempo de transmisión de los nodos de 15,147 ms.

P.Nbr	RX	Time (us)	Length	Type	Sec	End	Ack	req	PAN	compr	Sequence number	Dest PAN	Dest Address	Source Address	MAC payload	NWK Frame control field	NWK Dest Address	NWK Src Address	Broadcast Radius	Broadcast Seq num	LQI	FCS
6		+141216 +157890	18	DATA	0	0	0	0	1		0x00	0xC9FE	0xFFFF	0x1000	Ruptura	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x1c	182	OK
7		+1479 +159569	27	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x0001	T 0x0001	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x20	185	OK
8		+3656 +163228	27	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x0003	T 0x0003	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x20	185	OK
9		+1925 +165150	27	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x0002	T 0x0002	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x20	187	OK
10		+5259 +170409	27	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x0004	T 0x0004	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x20	174	OK
11		+2629 +173037	27	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x0005	T 0x0005	R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0xe1	0x20	206	OK
12		+8254737 +8427774	22	DATA	0	0	0	0	1		0x01	0xC9FE	0xFFFF	0x1000	Fin Ruptura	R10 0x1 1 1 0 0 0 0	0x204E	0x7652	0x70	0x74	179	OK

Figura 3.17: Captura de mensajes prueba 6.

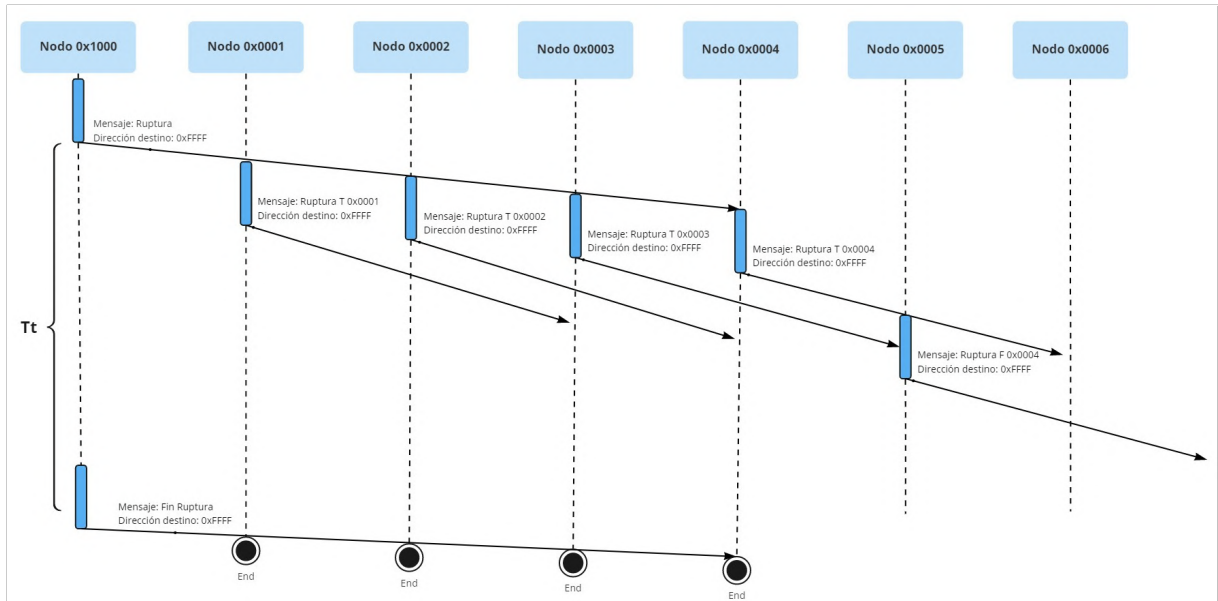


Figura 3.18: Transmisión de nodos prueba 6.

3.2.7 Prueba 7: Transmisión con el uso del algoritmo diseñado y detección de ruptura por cinco nodos

Finalmente, en esta prueba como se muestra en la Figura 3.19 y Figura 3.20, cinco nodos transmiten casi al mismo tiempo, demostrando que el algoritmo funciona para una cantidad N de nodos que detecten la ruptura o anomalía, evitando por completo la transmisión de cinco mensajes idénticos y retransmitiendo solo uno, cumpliendo con la finalidad de evitar la redundancia y optimizando los recursos disponibles, el tiempo de transmisión de los nodos es de 14,971 ms.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	NWK Frame control field	NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq. num	LQI	FCS
P.nbr. RX 1	+0	18	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xC9FE	0xFFFF	0x1000	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x19	179	OK
P.nbr. RX 2	+1367	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0002	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	158	OK
P.nbr. RX 3	+1243	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0004	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	171	OK
P.nbr. RX 4	+1456	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0001	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	187	OK
P.nbr. RX 5	+1606	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0003	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	155	OK
P.nbr. RX 6	+6853	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0005	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	174	OK
P.nbr. RX 7	+2446	27	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x06	0xC9FE	0xFFFF	0x0006	Ruptura	Type Version DR GA Sec R10 0x4 1 1 0 0 0 0	0x7470	0x7275	0x61	0x20	142	OK
P.nbr. RX 8	+8251924	22	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xC9FE	0xFFFF	0x1000	Fin Ruptura	Type Version DR GA Sec R10 0x1 1 1 0 0 0 0	0x206E	0x7552	0x70	0x74	182	OK

Figura 3.19: Captura de mensajes prueba 7.

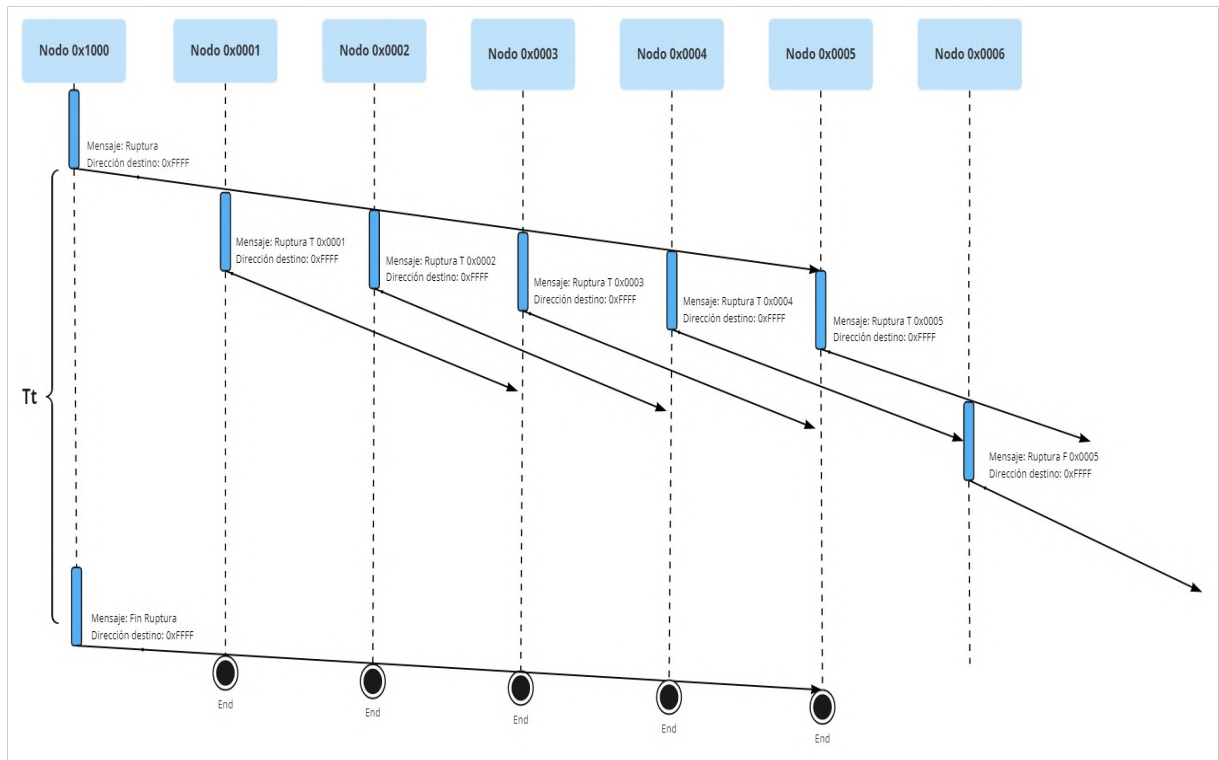


Figura 3.20: Transmisión de nodos prueba 7.

3.2.8 Problemas detectados

Al realizar las pruebas se puede observar cierto comportamiento impredecible de los nodos, los cuales envían sus mensajes con cierto retraso debido a que el canal se encuentra ocupado, o que recibió varias señales al mismo tiempo, cabe mencionar que este comportamiento no afecta al algoritmo mencionado, ya que el mensaje llegó correctamente en todas las pruebas realizadas, lo único que varía fue el orden en el cual decidieron enviar los primeros nodos. Este comportamiento puede ser causado por varios factores, como el nivel de batería, o el tiempo que tarda el sniffer y el nodo en procesar la información, cada aspecto pequeño puede influenciar en el orden final.

3.2.9 Comparación

Para determinar la mejora en la transmisión y eliminación de redundancia, es necesario comparar la cantidad de mensajes enviados y el tiempo de transmisión de las pruebas realizadas anteriormente Tabla 3.3.

Tabla 3.3: Comparación de las pruebas ejecutadas.

	Tramas transmitidas	Tiempo de transmisión (ms)
Prueba sin algoritmo diseñado	17	47,576
Pruebas con algoritmo diseñado		
Detección de ruptura por un nodo	6	9,930
Detección de ruptura por dos nodos	6	14,599
Detección de ruptura por tres nodos	7	20,301
Detección de ruptura por cuatro nodos	7	15,147
Detección de ruptura por cinco nodos	8	14,971
Promedio	6,8	14,989
Porcentaje de mejora	40 %	31,505 %

Existe una mejora del 40 % en el ahorro de la cantidad de tramas para los mismos nodos y una mejora del 31,505 % en el tiempo utilizado para la transmisión, estos resultados demuestran que el algoritmo ahorra un gran porcentaje de recursos y en el caso de existir muchos más nodos esta optimización sería crucial para alargar la vida útil de cada nodo y de la red con topología lineal.

3.2.10 Análisis de los resultados obtenidos

Las pruebas realizadas demuestran el funcionamiento del algoritmo para la eliminación de tramas repetidas en una topología lineal utilizando el estándar IEEE 802.15.4, se muestran los siguientes resultados:

- Se generó un ejemplo en el cual se produce redundancia, demostrando que el problema puede surgir debido a que varios nodos transmiten exactamente el mismo mensaje; además, estos mensajes circulan por toda la red de topología lineal, haciendo trabajar en exceso a los nodos y consumiendo recursos.
- El algoritmo desarrollado, diseñado e implementado con las características de la subcapa MAC y el estándar IEEE 802.15.4, puede eliminar tramas repetidas antes de que se produzcan entre los nodos involucrados, permitiendo que solo llegue un mensaje de ruptura, y evitando datos incensarios en la red.

- Cuando el sensor se encuentra procesando su trama y esperando que el canal se encuentre libre, el orden de transmisión varía, esto ocurre debido a que otro nodo puede enviar su trama con un tiempo de diferencia, cabe mencionar que el algoritmo no es afectado ya que no depende de esto.
- La herramienta de monitoreo Sniffer, presenta varias limitaciones de captura de tramas, debido a su sensibilidad para detectar dos tramas enviadas casi simultáneamente, causando que no aparezca una trama pero si su retransmisión, para demostrar que las tramas si fueron enviadas se utilizó un segundo Sniffer, el cual ayudo a detectar las tramas que el primero no pudo detectar.
- Con el uso del algoritmo diseñado se obtuvo una mejora del 40% para la cantidad de tramas y del 31,505% para la cantidad de tiempo de transmisión, lo que representa un ahorro significativo en el uso de los recursos disponibles de la red; además, puede ser aprovechado para alargar su vida útil de los nodos, permitiendo que trabajen de una forma mucho más óptima.
- La transmisión fue más rápida con respecto al caso en el que el algoritmo no fue implementado, obteniendo una gran mejora de consumo de energía y mínima cantidad de mensajes necesarios.

3.3 CONCLUSIONES

El desarrollo y creación del algoritmo con el uso del estándar IEEE 802.15.4 a nivel de capa de enlace para la eliminación de tramas repetidas en una topología lineal permite crear redes óptimas reduciendo el gasto de los recursos utilizados por los nodos al impedir el envío de mensajes innecesarios a lo largo de todos los nodos transceptores.

El algoritmo de eliminación de tramas repetidas dentro de una topología lineal multisalto y de comunicación unidireccional permite realizar un proceso en el cual no es posible el envío de mensajes con los mismos datos después de que se generó una alerta, reduciendo a uno, la cantidad de mensajes que llegan al final.

El prototipo creado con el estándar IEEE 802.15.4 y el microcontrolador ATmega256RFR2 en conjunto con el software Microchip Studio Atmel, permite ejecutar el algoritmo de la manera más eficiente, aportando así una mayor facilidad al momento de cargar los códigos a los nodos, y generando la ruptura que inicia la comunicación dentro de la topología lineal.

Los transceptores o nodos incorporados, permitieron ejecutar un algoritmo totalmente compatible con las características de IEEE 802.15.4, generando la posibilidad de manejar la redundancia a nivel de capa de enlace, así como también todas las variables para una comunicación estable.

Los datos obtenidos en las pruebas realizadas reflejan una corrección total de la redundancia demostrando una mejora del 40 % en el ahorro de la cantidad de mensajes y del 31,505 % en el ahorro de la cantidad del tiempo utilizado para comunicar un único mensaje, evitando así el consumo excesivo de recursos como ancho de banda y batería.

3.4 RECOMENDACIONES

En futuras aplicaciones del algoritmo se recomienda implementar una comunicación bidireccional, con la finalidad de comunicar a los extremos de la red ya que existen anomalías que deben ser solucionadas, de esta manera el administrador de la red estará informado de los eventos y tomara las medidas necesarias para la solución del caso.

Es recomendable que, para el correcto análisis y un adecuado monitoreo de la red, es necesario el uso de un dispositivo con mejores características y una mayor sensibilidad para evitar la pérdida de detección de tramas.

Es muy recomendable que antes de realizar todas las pruebas del algoritmo, se deba revisar el nivel de carga de la fuente de alimentación de los nodos, ya que afecta a la potencia y tiempo de procesamiento de los sensores.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. O. Sosa, *Contribuciones al establecimiento de una red global de sensores inalámbricos interconectados*, 2011.
- [2] L. E. Palafox Maestre, *Diseminación de datos en redes inalámbricas de sensores redundantes*, 2009.
- [3] A. B. Corral I., *Diseño e implementación de un entorno de simulación para redes de sensores inalámbricos [Ingeniería de Telecomunicaciones]*, Universidad Politécnica de Cartagena, 2005.
- [4] Atmel, *Microcontroller with Low Power 2.4GHz Transceiver for Zigbee and IEEE 802.15.4*, 2016.
- [5] S. E. Castillo Rodríguez, *Análisis de tecnologías WiFi y ZigBee que optimice las comunicaciones inalámbricas para el control de temperatura de un invernadero*, Escuela Superior Politécnica de Chimborazo, Riobamba, 2012.
- [6] F. D. Cali Reyes, *IMPLEMENTACIÓN DEL ALGORITMO DE PROTOCOLO DE DIRECCIONAMIENTO PARA REDES DE SENSORES INALÁMBRICOS CON EL ESTANDAR IEEE 802.15.4.*, ESCUELA POLITÉCNICA NACIONAL, Quito, 2018. dirección: <http://bibdigital.epn.edu.ec/handle/15000/19061>.
- [7] N. Aakvaag y J.-E. Frey, *Redes de Sensores Inalámbricos*, Revista ABB, vol. 2, pp. 39-42, 2006.
- [8] F. Ortiz Tapia, *Redes de sensores inalámbricos*, 2006.
- [9] A. Koubaa, M. Alves, y E. Tovar, *IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview*», jul. 2005. dirección: https://www.researchgate.net/publication/253418115_IEEE_802154_for_Wireless_Sensor_Networks_A_Technical_Overview.

- [10] J. Izquierdo Álvarez, *Mitigación de ataques en la Capa de Enlace de Datos mediante Detección y Monitoreo de la red.*, INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY, NL., 2004. dirección: <http://hdl.handle.net/11285/571900>.
- [11] L. E. Vera Sánchez, *IMPLEMENTACIÓN DE UN PROTOTIPO DE RED DE SENSORES INALÁMBRICOS EN TOPOLOGÍA LINEAL PARA LA DETECCIÓN DE EVENTOS UTILIZANDO EL ESTANDAR IEEE 802.15.4.*, ESCUELA POLITÉCNICA NACIONAL, Quito, 2018. dirección: <http://bibdigital.epn.edu.ec/handle/15000/22645>.
- [12] Atmel Corporation, *Programmers and Debuggers Atmel-ICE USER GUIDE*, Technology Drive, San Jose, CA 95110 USA, 2018. dirección: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-ICE_UserGuide.pdf.
- [13] Texas Instruments, *CC2531 USB Hardware User's Guide*, Texas Instruments Incorporated, Dallas, Texas, 2006. dirección: <https://www.ti.com/lit/ug/swru221a/swru221a.pdf?ts=1672332503189>.

5 ANEXOS

5.1 ANEXO I: REPOSITORIO Y BASE DE DATOS DEL PROYECTO

Los códigos generados en este proyecto se encuentran cargados en el siguiente enlace:

<https://github.com/IsidroN/Trabajo-de-Integracion-Curricular.git>

5.2 ANEXO II: MANUAL ATMEL 256RFR2

5.2.1 Atmel software Framework

Una vez descargado e instalado Atmel Studio a través de su página oficial, se abre la opción:
Tools >Extension Manager

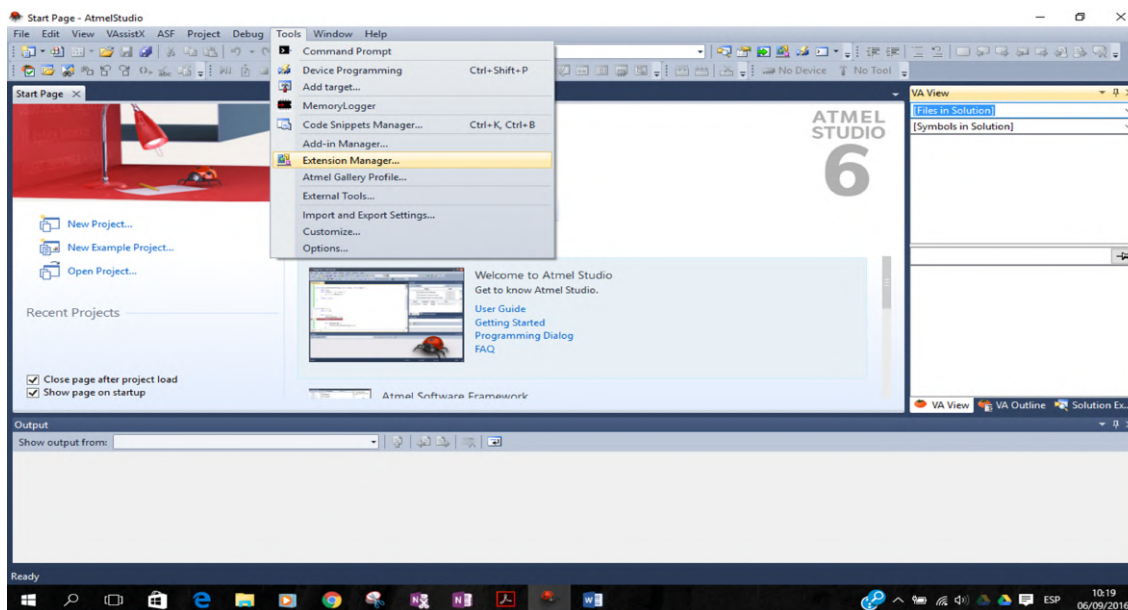


Figura 5.1: Paso 1.

Se selecciona “Atmel software Framework”

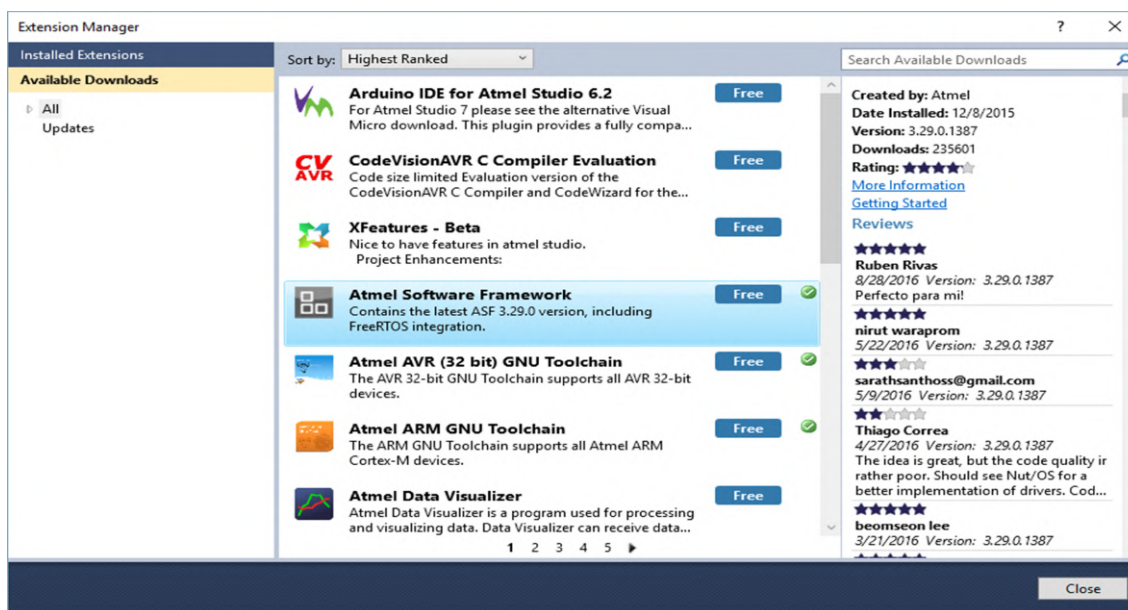


Figura 5.2: Paso 2.

En la siguiente ventana dar click en “next”

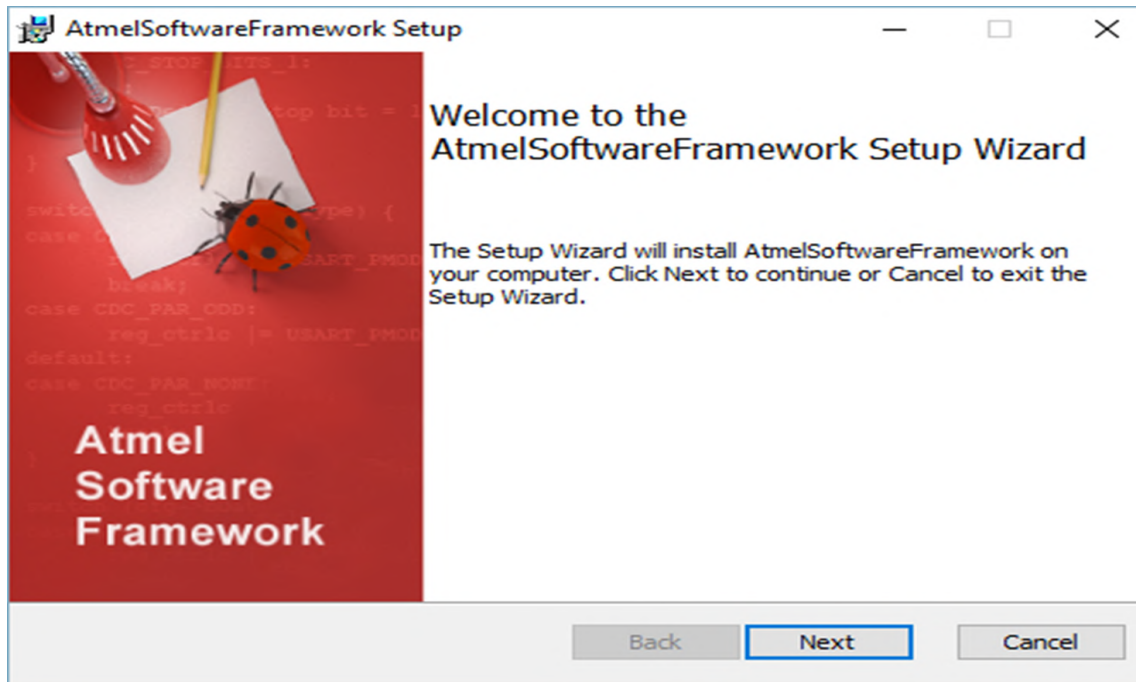


Figura 5.3: Paso 3.

Dar varias veces clic en “next”

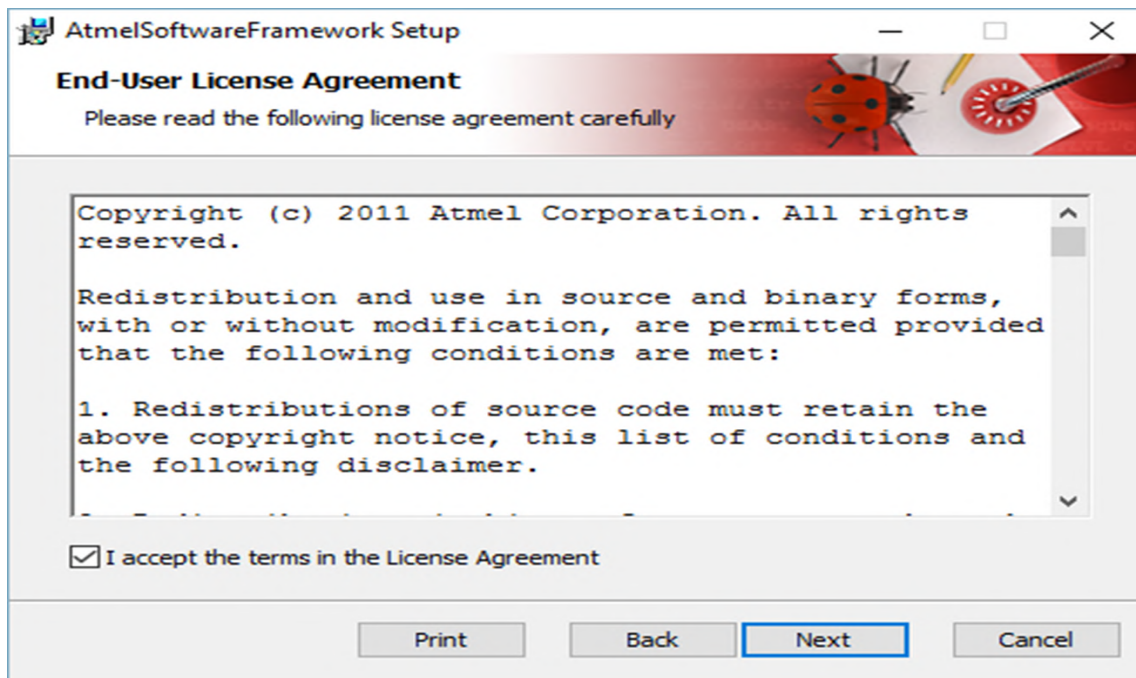


Figura 5.4: Paso 4.

Dar clic en “Install” y esperar que termine el proceso de instalación, para finalmente dar clic en “Finish”

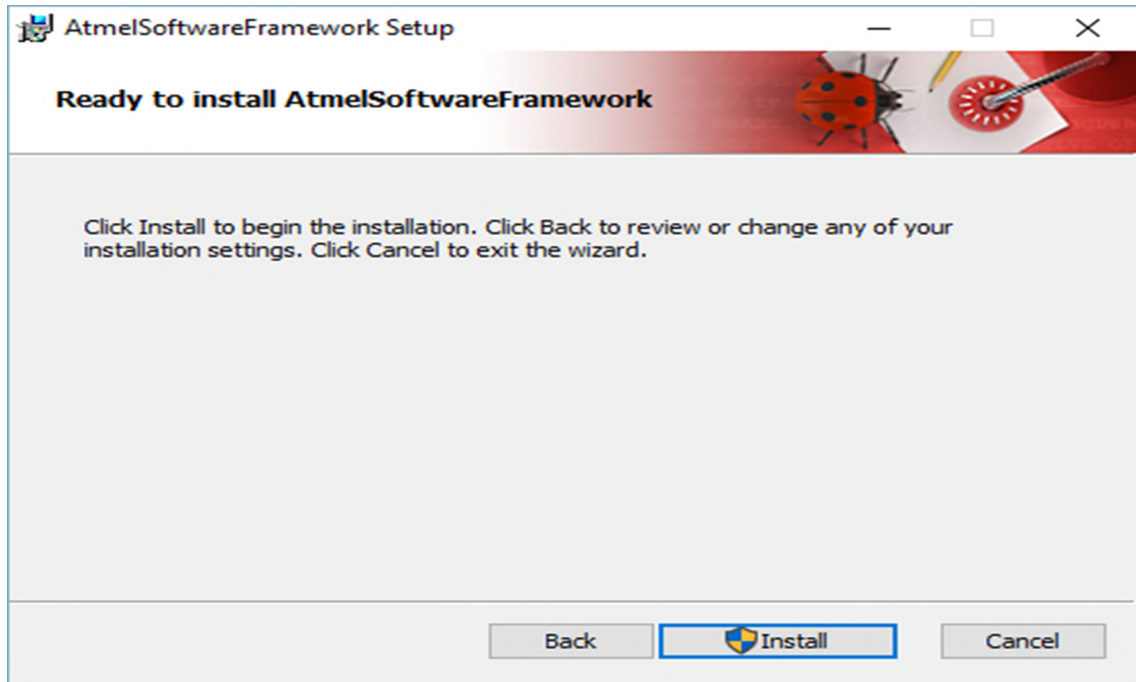


Figura 5.5: Paso 5.

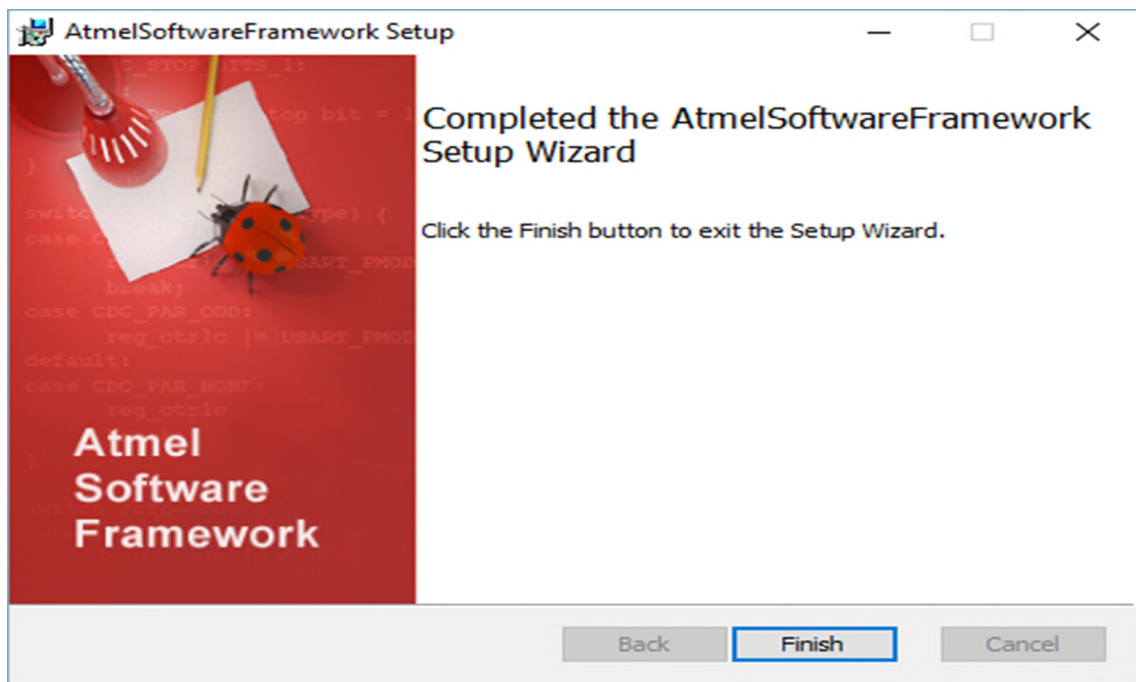


Figura 5.6: Paso 6.

5.2.2 II. Instalación Wireless composer 6.2

Una vez descargado e instalado Atmel Studio a través de su página oficial, se abre la opción: Tools >Extension Manager para luego dar clic en Download en Wireless Composer 6.2

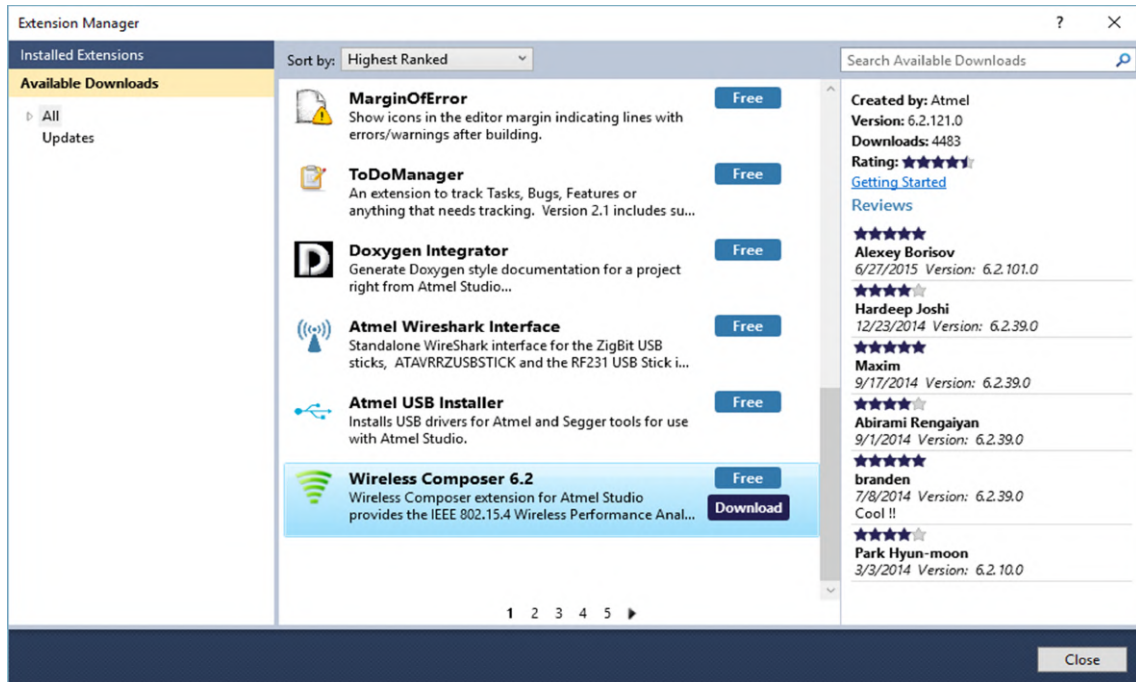


Figura 5.7: Paso 7.

Es necesario loguearse o crear una cuenta de atmel.

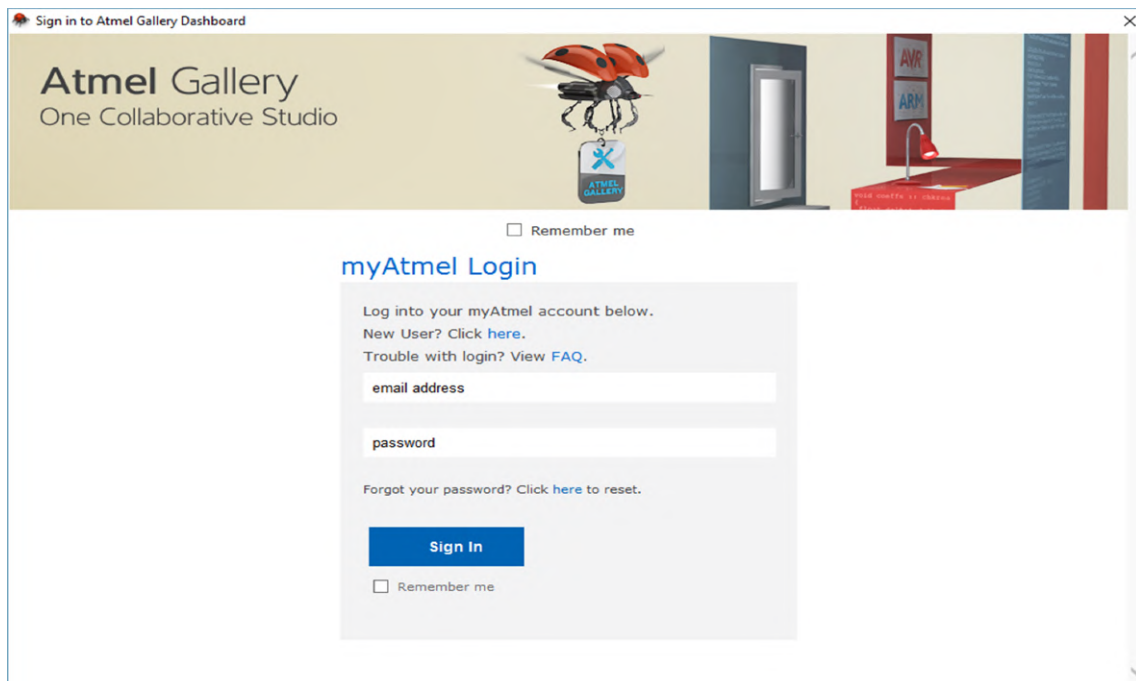


Figura 5.8: Paso 8.

Sign in to Atmel Gallery Dashboard

New User Registration

In order to complete setting up your account, please accept the Terms and Conditions of Service for using Atmel Gallery.

First Name: Saul
 Last Name: Ruiz
 Display Name: Saul Ruiz
 Email Address: saulruiz13@gmail.com

Country: Ecuador
 State: Pichincha
 City: Quito
 Zip Code:

Atmel Gallery Terms and Conditions of Service ("Terms")
 Last updated: November 2, 2012

In these Terms, "we," "us," "its" and "our" refer to the applicable Atmel company specified in Section 17.8 below (the "Company" or "Atmel"), and "you" and "your" refer to you. Before using the gallery.atmel.com website, any of its subdomains or the Atmel Gallery features within the Atmel Studio software (collectively, together with the services made available on or through such websites and features, the "Atmel Gallery"), please read these Terms carefully. These Terms are a binding legal agreement between you and Atmel regarding your use of the Atmel Gallery.

Accept Terms and Conditions.

SUBMIT

Figura 5.9: Paso 9.

Dar clic en "Install"

Download and Install

Install Wireless Composer 6.2

This extension does not contain a digital signature.

Copyright (c) 2011 Atmel Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Each extension is licensed to you under a License Agreement by its owner, not Microsoft. Microsoft does not guarantee the extension or purport to grant rights to it.

By clicking "Install", you agree with the above license terms (if any).

Install **Cancel**

Figura 5.10: Paso 10.