

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN ALGORITMO PARA LA OBTENCIÓN DE LA TABLA DE ENRUTAMIENTO, EN REDES CON TOPOLOGÍA TIPO ÁRBOL EMPLEANDO EL ESTÁNDAR IEEE 802.15.4

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

BRANDO STEVEN CURIPALLO PERALTA

DIRECTOR: ING. CARLOS ROBERTO EGAS ACOSTA, MSc.

Quito, mayo 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Brando Steven Curipallo Peralta, bajo mi supervisión.

Ing. Carlos Roberto Egas Acosta, MSc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

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

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

Brando Steven Curipallo Peralta

DEDICATORIA

A todas las personas que de alguna u otra forma fueron parte de este logro.

AGRADECIMIENTO

Agradezco a mis padres, Hilda y Luis, quienes me han brindado su amor y apoyo incondicional a lo largo de toda mi vida, en especial en mi etapa universitaria. Mediante su sacrificio he podido llegar a este punto tan importante de mi vida por lo que este logro también es de ustedes.

A mis hermanos Alex y Estefania, quienes han estado en todo momento dándome palabras de ánimo para no rendirme y seguir adelante.

Al MSc. Carlos Roberto Egas Acosta por la oportunidad de realizar el presente trabajo de titulación baso su guía.

A la Escuela Politécnica Nacional, a todos los docentes que fueron parte de mi formación académica en la carrera de Ingeniería en Electrónica y Redes de Información.

A todos los amigos que pude hacer durante esta etapa de mi vida.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS	XIII
ÍNDICE DE CÓDIGOS.....	XIV
RESUMEN.....	XVI
ABSTRACT.....	XVII
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE.....	1
1.3 MARCO TEÓRICO	4
1.3.1 REDES INALÁMBRICAS DE SENSORES	4
1.3.1.1 Definición.....	4
1.3.1.2 Elementos	4
1.3.1.3 Características.....	5
1.3.1.4 Aplicaciones	6
1.3.2 ESTÁNDAR IEEE 802.15.4	7
1.3.2.1 Características Generales	7
1.3.2.2 Tipos de dispositivos	7
1.3.2.3 Topologías.....	8
1.3.2.4 Arquitectura	10
1.3.2.4.1 Capa Física	10
1.3.2.4.2 Subcapa MAC (Medium Access Control).....	13

1.3.3	TRANSCEPTOR ATZB-256RFR2-XPRO	20
1.3.3.1	Características.....	21
1.3.3.2	Programación	22
1.3.3.3	Microcontrolador ATmega256RFR2	23
1.3.4	ENTORNO DE DESARROLLO ATMEL.....	23
1.3.4.1	Arquitectura	24
1.3.4.1.1	Capa PAL.....	24
1.3.4.1.2	Capa TAL.....	24
1.3.4.1.3	Capa MCL.....	25
1.3.4.1.4	Funcionamiento de las capas PAL, TAL, MCL.....	25
2.	METODOLOGÍA.....	26
2.1.	DESCRIPCIÓN DEL ALGORITMO.....	26
2.1.1	ALGORITMO PARA LA ASIGNACIÓN DE DIRECCIONES.....	26
2.1.1.1	Algoritmo de asignación de direcciones por orden de llegada de oferta de asignación.....	30
2.1.1.2	Algoritmo de asignación de direcciones por nivel de potencia de la trama recibida	34
2.1.1.3	Algoritmo de asignación de direcciones por número de saltos.....	37
2.1.2	ALGORITMO PARA LA COMUNICACIÓN DE EXTREMO A EXTREMO ...	40
2.2.	CODIFICACIÓN DEL ALGORITMO.....	42
2.2.1	Codificación del algoritmo usando el criterio de selección de ruta por orden de llegada de oferta de asignación	49
2.2.2	Codificación del algoritmo usando el criterio de selección de ruta por nivel de potencia de la trama recibida	55
2.2.3	Codificación del algoritmo usando el criterio de selección de ruta por número de saltos	59
2.2.4	Codificación del algoritmo de comunicación de extremo a extremo	64
3.	RESULTADOS Y DISCUSIÓN	67
3.1.	PRUEBA 1	68
3.2.	PRUEBA 2.....	72

3.3.	PRUEBA 3.....	76
3.4.	PRUEBA 4.....	80
3.5.	PRUEBA 5.....	84
3.6.	PRUEBA 6.....	88
3.7.	PRUEBA 7.....	92
3.8.	PRUEBA 8.....	98
3.9.	PRUEBA 9.....	105
3.10.	PRUEBA 10	112
3.11.	ANÁLISIS DE RESULTADOS OBTENIDOS	115
4.	CONCLUSIONES Y RECOMENDACIONES	119
4.1.	CONCLUSIONES.....	119
4.2.	RECOMENDACIONES.....	121
5.	REFERENCIAS BIBLIOGRÁFICAS.....	122
	ANEXOS.....	125

ÍNDICE DE FIGURAS

Figura 1.1 Prototipo de red de topología tipo árbol.....	2
Figura 1.2. Elementos de hardware de un nodo sensor	4
Figura 1.3. Topología estrella.....	8
Figura 1.4. Topología árbol	9
Figura 1.5. Topología malla.....	10
Figura 1.6. Capas en las que trabaja el estándar IEEE 802.15.4	10
Figura 1.7. Estructura de canales del estándar IEEE 802.15.4.....	11
Figura 1.8. Formato de la PPDU.	13
Figura 1.9. Formato de la trama de datos.	14
Figura 1.10. Formato de la trama beacon.	15
Figura 1.11. Formato de la trama ACK.....	15
Figura 1.12. Formato de la trama de comandos MAC.	16
Figura 1.13. Estructura de la supertrama.	17
Figura 1.14. Comunicación con modo beacon habilitado.	19
Figura 1.15. Comunicación con modo beacon deshabilitado.....	19
Figura 1.16. Comunicación con modo beacon habilitado.	19
Figura 1.17. Comunicación con modo beacon deshabilitado.....	20
Figura 1.18. Transceptor ATZB-256RFR2-XPRO.....	20
Figura 1.19. Descripción general del transceptor.	21
Figura 1.20. Pines de los LEDs.....	21
Figura 1.21. Pines de los botones.....	22
Figura 1.22. Programador ATMEL-ICE.	23
Figura 2.1. Topología previa al proceso de direccionamiento.....	26
Figura 2.2. Topología tipo árbol con 2 niveles.	28
Figura 2.3. Topología tipo árbol con 3 niveles.	29
Figura 2.4. Diagrama de flujo del algoritmo de asignación de direcciones por orden de llegada de la oferta de asignación.....	33

Figura 2.5. Diagrama de flujo del algoritmo de asignación de direcciones por nivel de potencia de la trama recibida.	36
Figura 2.6. Diagrama de flujo del algoritmo de asignación de direcciones por número de saltos.	39
Figura 2.7. Diagrama de flujo del algoritmo para la comunicación de extremo a extremo.	41
Figura 2.8. Solution Explorer con los archivos de configuración generados.	43
Figura 3.1. Ubicación del transceptor ATZB-256RFR2-XPRO en el cono.	68
Figura 3.2. Ubicación del transceptor RCB256RFR2 en el cono.	68
Figura 3.3. Sniffer conectado a la laptop.	68
Figura 3.4. Ubicación de los nodos antes de la asignación de direcciones fijas.	69
Figura 3.5. Direcciones aleatorias asignadas a los nodos.	69
Figura 3.6. Asignación de direcciones del Gateway.	70
Figura 3.7. Asignación de direcciones de los nodos asignadores.....	71
Figura 3.8. Topología final.....	71
Figura 3.9. Ubicación de los nodos antes de la asignación de direcciones fijas.	72
Figura 3.10. Direcciones aleatorias asignadas a los nodos.	73
Figura 3.11. Asignación de direcciones del Gateway.	74
Figura 3.12. Asignación de direcciones de los nodos asignadores.....	75
Figura 3.13. Topología final.....	75
Figura 3.14. Ubicación de los nodos antes de la asignación de direcciones fijas.	76
Figura 3.15. Direcciones aleatorias asignadas a los nodos.	76
Figura 3.16. Asignación de direcciones del Gateway.	77
Figura 3.17. Asignación de direcciones de los nodos asignadores.....	78
Figura 3.18. Topología con la dirección del nodo que se encendió.	79
Figura 3.19. Dirección aleatoria del nodo que se encendió.	79
Figura 3.20. Asignación de dirección al nodo caído.	79
Figura 3.21. Topología final.....	80
Figura 3.22. Ubicación de los nodos antes de la asignación de direcciones fijas.	80

Figura 3.23. Direcciones aleatorias asignadas a los nodos.	81
Figura 3.24. Asignación de direcciones del Gateway.	82
Figura 3.25. Asignación de direcciones de los nodos asignadores.	83
Figura 3.26. Topología final.	83
Figura 3.27. Ubicación de los nodos antes de la asignación de direcciones fijas.	84
Figura 3.28. Direcciones aleatorias asignadas a los nodos.	84
Figura 3.29. Asignación de direcciones del Gateway.	85
Figura 3.30. Asignación de direcciones de los nodos asignadores.	87
Figura 3.31. Topología final.	87
Figura 3.32. Ubicación de los nodos antes de la asignación de direcciones fijas.	88
Figura 3.33. Direcciones aleatorias asignadas a los nodos.	88
Figura 3.34. Asignación de direcciones del Gateway.	89
Figura 3.35. Asignación de direcciones de los nodos asignadores.	90
Figura 3.36. Topología con la dirección del nodo que se encendió.	90
Figura 3.37. Dirección aleatoria del nodo que se encendió.	90
Figura 3.38. Asignación de dirección al nodo caído.	91
Figura 3.39. Topología final.	91
Figura 3.40. Ubicación de los nodos antes de la asignación de direcciones fijas.	92
Figura 3.41. Direcciones aleatorias asignadas a los nodos.	93
Figura 3.42. Asignación de direcciones del Gateway.	94
Figura 3.43. Mensaje con la letra “S”, acompañada del valor del contador de saltos.	94
Figura 3.44. Asignación de direcciones de los nodos asignadores.	95
Figura 3.45. Mensaje con la letra “S”, acompañada del valor del contador de saltos.	96
Figura 3.46. Topología con la dirección del nodo que se encendió.	96
Figura 3.47. Dirección aleatoria del nodo que se encendió.	96
Figura 3.48. Asignación de dirección al nodo para probar escenario.	97
Figura 3.49. Mensaje con la letra “S”, acompañada del valor del contador de saltos.	97
Figura 3.50. Topología final.	98
Figura 3.51. Ubicación de los nodos antes de la asignación de direcciones fijas.	99

Figura 3.52. Direcciones aleatorias asignadas a los nodos.	99
Figura 3.53. Asignación de direcciones del Gateway.	100
Figura 3.54. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	101
Figura 3.55. Asignación de direcciones de los nodos asignadores.....	102
Figura 3.56. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	102
Figura 3.57. Topología con la dirección del nodo que se encendió.	103
Figura 3.58. Dirección aleatoria del nodo que se encendió.	103
Figura 3.59. Asignación de dirección al nodo para probar escenario.....	104
Figura 3.60. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	104
Figura 3.61. Topología final.....	105
Figura 3.62. Ubicación de los nodos antes de la asignación de direcciones fijas.	106
Figura 3.63. Direcciones aleatorias asignadas a los nodos.	106
Figura 3.64. Asignación de direcciones del Gateway.	107
Figura 3.65. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	107
Figura 3.66. Asignación de direcciones de los nodos asignadores.....	108
Figura 3.67. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	108
Figura 3.68. Topología con la dirección del nodo que se encendió.	109
Figura 3.69. Dirección aleatoria del nodo que se encendió.	109
Figura 3.70. Asignación de dirección al nodo para probar escenario.....	110
Figura 3.71. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	110
Figura 3.72. Topología con la dirección del nodo que se encendió.	110
Figura 3.73. Dirección aleatoria del nodo que se encendió.	111
Figura 3.74. Asignación de dirección al nodo caído.	111
Figura 3.75. Mensaje con la letra “S”, acompañada del valor del contador de saltos. ...	111
Figura 3.76. Topología final.....	112
Figura 3.77. Direcciones aleatorias asignadas a los nodos.	113
Figura 3.78. Asignación de direcciones del Gateway.	113
Figura 3.79. Asignación de direcciones de los nodos asignadores.....	114
Figura 3.80. Topología final.....	114

Figura 3.81. Proceso de comunicación del nodo al Gateway.	115
Figura 3.82. Topología con la ruta de comunicación del nodo al Gateway.	115
Figura 3.83. Valores enviados por los nodos asignadores a los nodos.	116
Figura 3.84. Valores enviados por los nodos asignadores a los nodos.	116
Figura 3.85. Valores enviados por los nodos asignadores a los nodos.	116

ÍNDICE DE TABLAS

Tabla 1.1. Parámetros de modulación del estándar IEEE 802.15.4	12
Tabla 2.1. Cantidad de bits usados para la dirección del coordinador.	27
Tabla 2.2. Cantidad de bits usados para la dirección del coordinador.	28
Tabla 2.3. Cantidad de bits usados para direccionar los nodos del segundo nivel.....	28
Tabla 2.4. Cantidad de bits usados para direccionar los nodos del tercer nivel (Nodo 1).	30
Tabla 2.5. Cantidad de bits usados para direccionar los nodos del tercer nivel (Nodo 2)..	30
Tabla 2.6. Elementos necesarios para generar y comprobar el funcionamiento del código.	42
Tabla 3.1. Tiempos obtenidos en el proceso de asignación de direcciones.	117

ÍNDICE DE CÓDIGOS

Código 2.1. Estructuras creadas en <code>usr_wireless.h</code>	44
Código 2.2. Declaración de librerías.	44
Código 2.3. Declaración de directivas y constantes.	44
Código 2.4. Declaración de variables enteras con y sin signo.....	45
Código 2.5. Declaración de variables booleanas.....	45
Código 2.6. Declaración de variables tipo char.	46
Código 2.7. Declaración de variables booleanas y tipo char.	46
Código 2.8. Declaración de variables enteras con y sin signo.....	46
Código 2.9. Función para enviar información a los nodos para que definan su dirección fija.....	48
Código 2.10. Función para visualizar la dirección aleatoria de los nodos.....	48
Código 2.11. Función para asignar un estado a los nodos.....	49
Código 2.12. Función para inicio del proceso de direccionamiento por orden de llegada.	49
Código 2.13. Función para envío de mensaje en broadcast.....	50
Código 2.14. Procedimiento inicial al recibir una trama.	50
Código 2.15. Procedimiento de los nodos al recibir una trama con dirección broadcast..	50
Código 2.16. Procedimiento en caso de que sea un nodo asignador.	50
Código 2.17. Procedimiento en caso de que sea más de un nodo asignador.	51
Código 2.18. Procedimiento del nodo asignador al recibir respuestas de los nodos.	51
Código 2.19. Procedimiento del nodo asignador al recibir respuestas de los nodos.	52
Código 2.20. Procedimiento de los nodos para calcular y definir su nueva dirección fija.	53
Código 2.21. Procedimiento del nodo al simular un escenario de enlace ruidoso.	53
Código 2.22. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.	54
Código 2.23. Procedimiento del nodo al simular un escenario de nodo caído.....	54
Código 2.24. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.....	55

Código 2.25. Función para inicio del proceso de direccionamiento por nivel de potencia.	55
Código 2.26. Función para envío de mensaje en broadcast.....	55
Código 2.27. Procedimiento de los nodos al recibir una trama con dirección broadcast..	56
Código 2.28. Procedimiento en caso de que sea un nodo asignador.	56
Código 2.29. Procedimiento en caso de que sea más de un nodo asignador.	57
Código 2.30. Procedimiento del nodo asignador al recibir respuestas de los nodos.	57
Código 2.31. Procedimiento del nodo al simular un escenario de enlace ruidoso.	58
Código 2.32. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.	58
Código 2.33. Procedimiento del nodo al simular un escenario de nodo caído.....	59
Código 2.34. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.....	59
Código 2.35. Función para inicio del proceso de direccionamiento por número de saltos.	60
Código 2.36. Función para envío de mensaje en broadcast.....	60
Código 2.37. Procedimiento de los nodos al recibir una trama con dirección broadcast..	60
Código 2.38. Procedimiento en caso de que sea un nodo asignador.	61
Código 2.39. Procedimiento en caso de que sea más de un nodo asignador.	61
Código 2.40. Procedimiento del nodo asignador al recibir respuestas de los nodos.	62
Código 2.41. Procedimiento del nodo al simular un escenario de enlace ruidoso.	62
Código 2.42. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.	63
Código 2.43. Procedimiento del nodo al simular un escenario de nodo caído.....	63
Código 2.44. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.....	64
Código 2.45. Configuración de los nodos para el proceso de comunicación.	64
Código 2.46. Proceso de transmisión en la comunicación.	65
Código 2.47. Proceso de recepción en la comunicación.	66
Código 2.48. Transmisión a un nodo al cual le asignó la dirección.	66

RESUMEN

El presente trabajo de titulación tiene como objetivo desarrollar un algoritmo que permita obtener la tabla de enrutamiento de los nodos que forman parte de una red WSN. Esto se realizará utilizando la información que se genera durante el proceso de asignación de direcciones de forma jerárquica haciendo uso del estándar IEEE 802.15.4, cabe mencionar que durante este proceso se simulará varios escenarios entre los que se tiene: escenario ideal (sin fallos), escenario con enlaces ruidosos y escenario con nodos caídos.

En el primer capítulo se presenta la introducción, los objetivos, el alcance y el marco teórico, dentro de este último se hace un repaso de las redes inalámbricas de sensores (WSN), también se realiza un estudio general del estándar IEEE 802.15.4, además se indican las características del transceptor ATZB-256-RFR2 y del entorno de desarrollo Atmel Studio 7.0.

En el segundo capítulo se realiza la descripción del algoritmo para cada uno de los criterios de asignación de direcciones, dentro de esta sección se incluyen los diagramas de flujo de cada criterio, además se presenta la implementación del algoritmo, en esta sección se muestran las partes más importantes del código que se desarrolló.

En el tercer capítulo se presenta la descripción de las pruebas realizadas con las respectivas capturas obtenidas a través del capturador de tramas (sniffer) y las diferentes topologías tipo árbol obtenidas para cada criterio de asignación de direcciones, también se presenta el análisis de los resultados obtenidos en las distintas pruebas.

En el cuarto capítulo se presentan las conclusiones y recomendaciones.

Finalmente, en la sección de anexos se adjunta el código comentado y los distintos diagramas de flujo.

PALABRAS CLAVE: transceptor, ATZB-256RFR2-XPRO, Gateway, asignador, nodo, WSN, sniffer, IEEE 802.15.4, topología.

ABSTRACT

The objective of this degree work is to develop an algorithm that allows obtaining the routing table of the nodes that are part of a WSN network. This will be done using the information that is generated during the hierarchical address assignment process using the IEEE 802.15.4 standard. It is worth mentioning that during this process several scenarios will be simulated, including ideal scenario (without failures), scenario with noisy links and scenario with down nodes.

In the first chapter the introduction, the objectives, the scope and the theoretical framework are presented, within the latter a review of wireless sensor networks (WSN) is made, a general study of the IEEE 802.15.4 standard is also carried out, in addition, the characteristics of the ATZB-256-RFR2 transceiver and the Atmel Studio 7.0 development environment are indicated.

In the second chapter, the description of the algorithm is made for each of the criteria for requesting addresses, within this section the flow diagrams of each criterion are included, in addition the implementation of the algorithm is presented in this section are shown the most important parts of the code that was developed.

The third chapter presents the description of the tests carried out with the respective captures obtained through the frame capturer (sniffer) and the different tree-type topologies obtained for each address allocation criterion, as well as the analysis of the results obtained in the different tests.

The fourth chapter presents the conclusions and recommendations.

Finally, in the annexes section, the commented code and the different flowcharts are attached.

KEY WORDS: transceiver, ATZB-256RFR2-XPRO, Gateway, allocator, node, WSN, sniffer, IEEE 802.15.4, topology.

1. INTRODUCCIÓN

Los requisitos de cualquier solución WSN (Redes Inalámbricas de Sensores) siempre dependerán estrechamente de la aplicación concreta. En la actualidad la creación de nuevas aplicaciones orientadas a las WSN se dificulta debido a los retardos por procesamiento que se generan en los nodos sensores al definir una ruta de comunicación y recuperarse de nodos caídos o enlaces ruidosos. Por lo que es necesario proponer soluciones que permitan resolver el problema del retardo, disminuyendo el tiempo de procesamiento con el fin de que se puedan aplicar a redes multisalto a gran escala.

Además, se debe considerar que, al disminuir el tiempo de procesamiento de los nodos sensores para obtener la tabla de enrutamiento también se disminuye el consumo de energía, lo que permite alargar el tiempo de operación de los nodos.

Dicho esto, en el presente Trabajo de Titulación se plantea el desarrollo de un algoritmo para obtener la tabla de enrutamiento de los nodos que conforman una WSN haciendo uso de la información que se obtiene al realizar el proceso de asignación de direcciones de forma jerárquica basándose en el estándar IEEE 802.15.4, logrando de esta forma obtener una topología tipo árbol.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

- Implementar un algoritmo para la obtención de la tabla de enrutamiento, en redes con topología árbol empleando el estándar IEEE 802.15.4

Los objetivos específicos del Proyecto Técnico son:

- Analizar la base conceptual necesaria para para la implementación del algoritmo.
- Diseñar el algoritmo a nivel de capa enlace para obtener las tablas de enrutamiento.
- Implementar el algoritmo en los nodos.
- Analizar resultados obtenidos del funcionamiento del algoritmo.

1.2 ALCANCE

En el presente trabajo se desarrollará un algoritmo que permita la obtención de tablas de enrutamiento en el nodo. Para la obtención de la tabla de enrutamiento se utilizará la

información generada durante el proceso de asignación de direcciones utilizando direccionamiento jerárquico, el cual permitirá que todos los nodos que forman parte de la WSN se encuentren correctamente identificados. Luego de este proceso los nodos tendrán una tabla de enrutamiento, la cual almacenará información que les permita conectarse a cualquier nodo dentro de la red WSN, así como llegar al nodo Gateway el cual está conectado a la red Internet, por lo tanto, existirá conectividad de extremo a extremo entre los nodos sensores. Una vez realizado este proceso se obtendrá una topología como la que se observa en la Figura 1.1.

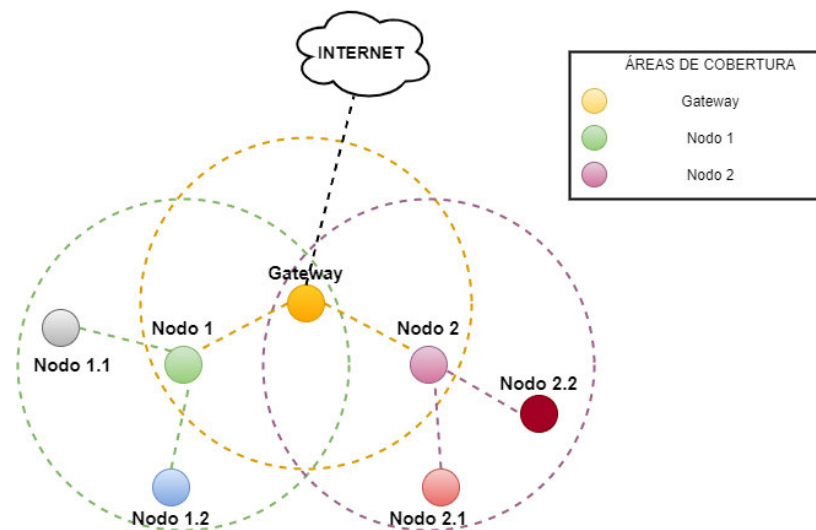


Figura 1.1 Prototipo de red de topología tipo árbol.

El proceso de asignación de direcciones para un nodo considera que dentro de la zona de cobertura de dicho nodo pueden existir uno o más nodos que requieran direcciones, la dirección asignada tendrá información que defina la ruta principal y las rutas de respaldo para alcanzar el Gateway u otro nodo que forme parte de la WSN. En el proceso de selección de la mejor ruta se considerarán tres criterios para seleccionar el nodo que permita la trama llegar a su destino.

Los criterios de selección para la ruta principal y las rutas secundarias son los siguientes:

- **Nivel de potencia de la trama recibida:** mientras mayor sea este valor, significa que el nodo transmisor se encuentra más cerca, por lo que el ruido afectará de menor manera a la señal recibida.
- **Número de saltos:** hace referencia a la cantidad de nodos por los cuales la información debe pasar para llegar a su destino, mientras menor sea este valor, los datos se pasarán por menos nodos retransmisores.

- **Orden de llegada de oferta de asignación:** esto quiere decir que el primer nodo del cual se reciba una oferta para asignar una dirección será la primera opción de ruta a seguir para el envío de información y en el orden que lleguen las ofertas será el orden de prioridad para el siguiente salto a realizar.

El prototipo de red a implementar para verificar el funcionamiento del algoritmo consta de al menos 7 nodos sensores con los cuales se implementarán varias topologías tipo árbol las cuales serán definidas en la fase de diseño del algoritmo. Los nodos sensores que se utilizarán en este trabajo de titulación serán los transceptores ATZB-256RFR2-XPRO los cuales trabajan con un microcontrolador ATmega256RFR2. Para la codificación del algoritmo propuesto se utiliza el lenguaje C/C++ en el entorno de programación Atmel Studio 7.

El algoritmo desarrollado se probará en escenarios con nodos caídos y enlaces ruidosos, permitiendo al nodo automáticamente buscar una nueva ruta para reestablecer la conectividad, haciendo uso de la información de la tabla de enrutamiento generada. Estos escenarios se tomarán en cuenta en el proceso de generación de la tabla de enrutamiento de tal manera de que el algoritmo propuesto sea robusto ante nodos caídos y enlaces ruidosos.

Los nodos trabajarán en el modo no ranurado con CSMA/CA y en la detección de nodos caídos y enlaces ruidosos se utilizará iACK (implicit ACK).

Las pruebas por realizar utilizan el prototipo a implementar, el cual está configurado para trabajar en una topología tipo árbol. Se comprobará el correcto funcionamiento del algoritmo, se medirán los tiempos de retardo en la generación de las tablas de enrutamiento y los tiempos de recuperación de fallos ante nodos caídos y enlaces ruidosos. Estas pruebas se realizarán en escenarios distintos. Uno de ellos será en un ambiente "ideal", es decir, en el que no existan fallos. El siguiente será en presencia de nodos caídos, lo cual se conseguirá apagando algún nodo. Finalmente, se conoce que un enlace ruidoso causa que una trama no sea recibida por un nodo, por lo tanto, cuando la trama que se pierde contiene información que contribuye a la obtención de la tabla de enrutamiento, como resultado se tiene una tabla de enrutamiento que no será confiable, por lo que el presente trabajo de titulación no tiene como objetivo simular el enlace ruidoso sino la consecuencia de este, es decir, mediante programación se hará que un nodo no procese dicha trama.

1.3 MARCO TEÓRICO

1.3.1 REDES INALÁMBRICAS DE SENSORES

Las redes inalámbricas de sensores (WSN, por sus siglas en inglés, Wireless Sensor Networks), son una tendencia tecnológica que cada día cuenta con más aplicaciones, las características de sus nodos, los protocolos que utilizan y la versatilidad de sus configuraciones, las convierten en una opción importante dentro del mundo tecnológico [1].

1.3.1.1 Definición

Las WSN están constituidas por sensores autónomos distribuidos espacialmente con el propósito de ser capaces de comunicarse entre sí con un mínimo de consumo de energía y entregando su colección de datos [2].

1.3.1.2 Elementos

Una red inalámbrica de sensores se compone de diferentes elementos, los cuales le permiten establecer una comunicación entre los nodos que la conforman, este tipo de redes se componen de nodos sensores, puerta de enlace, estación base y canal de transmisión inalámbrico, a continuación, se explican estos elementos [1]:

- **Nodos sensores:** son dispositivos electrónicos configurados a nivel de hardware y software para obtener información, están supeditados a la aplicación para la cual se hayan creado.

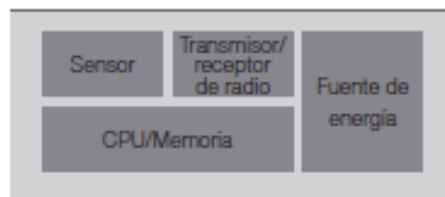


Figura 1.2. Elementos de hardware de un nodo sensor [3]

A continuación, se explican los elementos mencionados en la Figura 1.2 [1]:

- ✓ **Sensores:** son dispositivos de hardware que han sido creados para capturar información del entorno donde se ubiquen.
- ✓ **Transmisor / receptor de radio:** es un dispositivo de comunicación que permite enviar y recibir datos para comunicarse con otros dispositivos dentro de su rango de cobertura.

- ✓ **Procesador (CPU¹):** dispositivo que hace referencia al microcontrolador, el cual tiene las capacidades de procesamiento para implementar un protocolo de comunicación en la WSN.
 - ✓ **Memoria:** los nodos sensores también disponen de elementos de memoria que les permiten guardar la información que capturan, este tipo de memoria puede ser volátil o no volátil.
 - ✓ **Fuente de energía:** elemento que permite el funcionamiento eléctrico del nodo sensor, se refiere a las baterías o fuentes que alimentan los circuitos internos del nodo y les permite cumplir con el objetivo por el cual fueron diseñados.
- **Puerta de enlace:** dispositivo dentro de una WSN que permite recibir la información que han capturado los nodos sensores y enviarla a un centro de datos u otro dispositivo dentro de la misma red. En algunos casos la puerta de enlace puede ser otro nodo sensor o puede tratarse de un dispositivo de capa 3 (haciendo referencia al modelo OSI²).
 - **Estación base:** hace referencia al equipo que guardará la información que se capture mediante los nodos sensores, en algunos casos la estación base está conectada a una WSN directamente ya sea por algún medio guiado o inalámbrico.
 - **Canal de transmisión:** es el medio físico por el cual se transmite la información, de acuerdo con la naturaleza de la red con la que se está trabajando, se usará un medio no guiado (inalámbrico).

1.3.1.3 Características

A continuación, se detallan algunas características de las WSN [3] [4]:

- Las redes se pueden configurar en un estado de consumo mínimo mientras no tengan paquetes que transmitir, con lo cual se logra un bajo consumo de energía.
- En este tipo de redes al tener una alimentación autónoma por medio de paneles solares o baterías recargables, de esta forma se logra que los dispositivos funcionen por periodos de tiempo más largos.

¹ CPU, por sus siglas en inglés “Central Processing Unit”, se encarga de procesar todas las instrucciones del dispositivo.

² OSI, por sus siglas en inglés “Open Systems Interconnection”, es un modelo de referencia conformado por 7 capas cuyo objetivo es interconectar sistemas de diferentes fabricantes.

- Son redes de bajo costo, ya que los nodos no necesitan gran capacidad de procesamiento.
- Una WSN tiene capacidad de recuperación, es decir, si se avería un nodo, la red encontrará nuevas vías para encaminar los paquetes de datos.
- Los requisitos de cualquier solución WSN siempre dependerán estrechamente de la aplicación concreta.

1.3.1.4 Aplicaciones

Entre las aplicaciones de las WSN destacan las siguientes [5]:

- **Aplicaciones militares:**
 - ✓ Monitorización de fuerzas y equipos enemigos.
 - ✓ Vigilancia en el campo de batalla.
 - ✓ Reconocimiento del terreno.
 - ✓ Detección de ataques biológicos, químicos, nucleares, etc.
- **Aplicaciones medioambientales:**
 - ✓ Seguimiento de animales.
 - ✓ Monitorización de las condiciones ambientales en cultivos.
 - ✓ Detección de incendios forestales.
 - ✓ Detección de inundaciones.
 - ✓ Estudios de contaminación.
 - ✓ Estudios sísmicos.
- **Aplicaciones médicas:**
 - ✓ Monitorización de datos fisiológicos en pacientes.
 - ✓ Diagnóstico.
 - ✓ Administración de medicamentos.
 - ✓ Seguimiento de médicos y pacientes en hospitales.

1.3.2 ESTÁNDAR IEEE 802.15.4

El estándar IEEE 802.15.4 determina las comunicaciones en la capa física y en la subcapa de control de acceso al medio (MAC), se lo utiliza principalmente en redes inalámbricas de sensores ya que se caracterizan por la baja tasa de datos que transportan y el bajo consumo de energía requerido para su funcionamiento [2].

1.3.2.1 Características Generales

El estándar IEEE 802.15.4 tiene las siguientes características [6] [7]:

- Operan en las bandas de frecuencia de 868/915 MHz y 2.4 GHz.
- Trabaja con velocidades de datos de 250 kbps (2.4 GHz), 40 kbps (915 MHz) y 20 kbps (868 MHz).
- El método de acceso al canal es CSMA/CA³.
- Transmisión a baja potencia.
- Bajo uso de la batería.
- Puede trabajar en múltiples topologías: malla, árbol y estrella.
- Direccionamiento corto: 16 bits para identificar, determina una cantidad máxima de 65536 dispositivos (nodos).
- Alcance: hasta 75 metros (valor típico, depende del ambiente).

1.3.2.2 Tipos de dispositivos

El estándar IEEE 802.15.4 define dos tipos de dispositivos en la red [8] [9] [10] [11]:

- **Full Function Device (FFD):** es capaz de realizar todas las operaciones disponibles dentro del estándar y puede operar en tres modos diferentes:
 - ✓ *Coordinador PAN⁴:* se construye a partir de un FFD y es fundamental para formar una nueva red. Puede tener un conocimiento general de toda la red.

³ CSMA/CA, por sus siglas en inglés "Carrier Sense Multiple Access with Collision Avoidance" es un protocolo de acceso a redes en el que cada dispositivo anuncia su intención de transmitir para evitar colisiones.

⁴ PAN, por sus siglas en inglés "Personal Area Network" es un estándar de red para la comunicación entre dispositivos.

- ✓ *Coordinador*: realiza la sincronización de los dispositivos cuando se utilizan beacons⁵, debe pertenecer a una PAN ya que no es capaz de crearla.
- ✓ *Dispositivo final*: puede construirse a partir de cualquier dispositivo físico, solo maneja las comunicaciones y la transferencia de datos por sí mismo.
- **Reduced Function Device (RFD)**: está diseñado para ejecutar aplicaciones que son extremadamente simples. No necesitan enviar grandes cantidades de datos y solo se asocia con un único FFD a la vez. El propósito principal de un RFD es buscar una red disponible para transferir los datos, verificar si hay datos pendientes y enviar una solicitud al coordinador de la red para obtener los datos.

1.3.2.3 Topologías

El estándar IEEE 802.15.4 ofrece topologías de estrella, árbol y malla, las cuales se detallan a continuación [11] [12] [13]:

- **Topología tipo estrella**: consta de un coordinador colocado en el centro y varios dispositivos finales (nodos), como se muestra en la Figura 1.3. Cada nodo está conectado directamente con el coordinador central. En esta topología, los dispositivos finales solo pueden comunicarse con el coordinador y no con otros dispositivos finales. Cualquier intercambio de paquetes entre dispositivos finales solo puede ocurrir a través del coordinador.



Figura 1.3. Topología estrella [13]

- **Topología tipo árbol**: la red consta de un nodo raíz que es un coordinador PAN, varios coordinadores y dispositivos finales, como se muestra en la Figura 1.4. Todos los nodos están conectados en forma de árbol. Los nodos finales están conectados directamente al coordinador PAN y a los coordinadores como sus

⁵ Beacons: son tramas que contienen información sobre la red inalámbrica y son transmitidos periódicamente.

nodos secundarios. Tanto los enrutadores como el coordinador pueden tener hijos.

Un dispositivo final puede comunicarse con otro dispositivo final solo a través de su nodo principal y no existe una conexión directa entre los dispositivos finales. Los inconvenientes de la topología tipo árbol son que, si uno de los padres se desactiva, los hijos del padre desactivado no pueden comunicarse con otros dispositivos en la red.

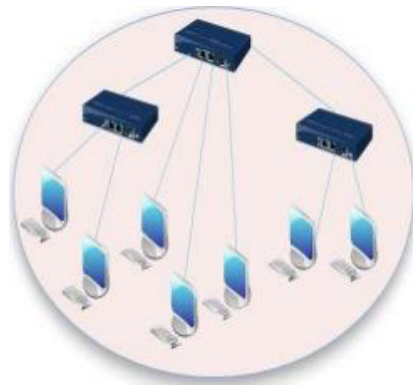


Figura 1.4. Topología árbol [13]

- **Topología tipo malla:** también se conoce como topología peer to peer. El coordinador puede enviar paquetes a cualquier nodo de la red. Si el nodo no está dentro del rango, el mensaje se enviará a un nodo vecino que luego lo reenviará al destino. La red de malla puede cubrir un rango más amplio usando solo una fracción de energía, además es capaz de crecer o reducirse simplemente agregando o quitando nodos.

Una topología de malla es autorreparable, es decir, durante la transmisión, si alguna de las rutas falla, el nodo encontrará una ruta alternativa al destino. Cualquier dispositivo de origen puede comunicarse con cualquier dispositivo de destino en la red. En comparación con la topología en estrella, la topología de malla requiere una mayor sobrecarga.

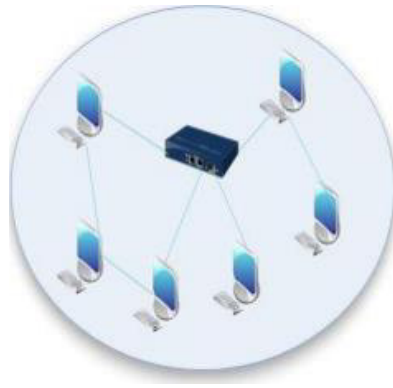


Figura 1.5. Topología malla [13]

1.3.2.4 Arquitectura

El estándar IEEE 802.15.4 define algunas especificaciones en las siguientes capas del modelo OSI [12]:

- Capa Física (PHY, Physical Layer).
- Subcapa MAC (Medium Access Control).

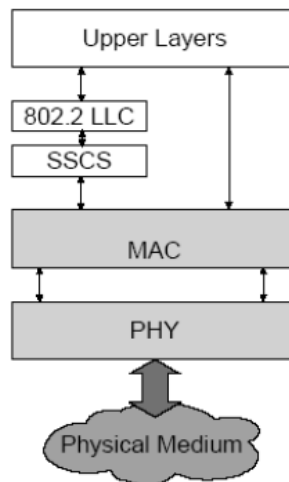


Figura 1.6. Capas en las que trabaja el estándar IEEE 802.15.4 [10]

1.3.2.4.1 Capa Física

El estándar IEEE 802.15.4 ofrece dos opciones de capa física. Ambas PHYs se basan en métodos de secuencia directa de espectro extendido (DSSS, Direct Sequence Spread Spectrum) que resultan en bajos costos de implementación en ICs (Integrated Circuits). La principal diferencia entre ambas PHYs radica en la banda de frecuencias en las que operan. La PHY de los 2.4 GHz, especifica operación en la banda ISM (Industrial, Scientific and Medical), que prácticamente está disponible a nivel mundial, mientras que

la PHY de los 868/915 MHz especifica operaciones en la banda de 865 MHz en Europa y 915 MHz en la banda ISM en Estados Unidos [15].

❖ Características

Las características de la capa física son las siguientes [6] [14]:

- *Activación y desactivación del transceptor.*
- *Detección de energía.*
- *Indicador de calidad de enlace (LQI, Link Quality Indicator).*
- *Detección de portadora (CCA, Clear Channel Assessment).*
- *Selección del canal de transmisión.*

❖ Canales

El estándar IEEE 802.15.4 define 27 canales los cuales están divididos en tres bandas de frecuencia como se observa en la Figura 1.7 [14]:

- La banda de frecuencia de 868 MHz soporta un solo canal entre los 868 y 868.6 MHz.
- La banda de frecuencia de los 915 MHz soporta 10 canales entre los 902 y 928 MHz.
- La banda de frecuencia de los 2.4 GHz soporta 16 canales.

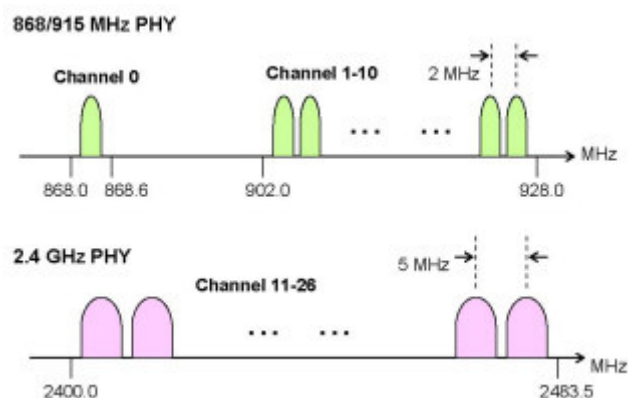


Figura 1.7. Estructura de canales del estándar IEEE 802.15.4 [9]

❖ Modulación

El estándar IEEE 802.15.4 para la transmisión de datos a través del medio utiliza la técnica DSSS (Direct Sequence Spread Spectrum – Espectro Ensanchado por

Secuencia Directa) que proporciona robustez contra interferencias. La PHY de 868/915 MHz emplea una técnica de modulación BPSK (Binary Phase Shift Keying – Modulación por desplazamiento de fase binaria), mientras que la PHY de 2.4 GHz usa una técnica de modulación O-QPSK (Offset Quadrature Phase Shift Keying – Modulación por desplazamiento de fase en cuadratura) [15]. En la Tabla 1.1 se muestran las características de modulación en capa física.

Tabla 1.1. Parámetros de modulación del estándar IEEE 802.15.4 [15]

PHY	Frequency band	Data parameters			Spreading parameters	
		Bit rate (kb/s)	Symbol rate (kbaud)	Modulation	Chip rate (Mchips/s)	Modulation
868/915	868.0–868.6 MHz	20	20	BPSK	0.3	BPSK
MHz PHY	902.0–928.0 MHz	40	40	BPSK	0.6	BPSK
2.4 GHz PHY	2.4–2.4835 GHz	250	62.5	16-ary orthogonal	2.0	O-QPSK

❖ *Formato de la PPDU (PHY Protocol Data Unit)*

El formato de la PDU de capa física está conformado por los campos que se muestran en la Figura 1.8. Dichos campos son los siguientes:

- **Preámbulo (Preamble):** constituido por 4 bytes es el encargado de permitir que el receptor adquiera y se sincronice con la señal entrante [10].
- **Delimitador de inicio de trama (Start of packet delimiter):** posee un tamaño de 1 byte y señala el fin del preámbulo [10].
- **Cabecera PHY (PHY header):** tiene un tamaño de 1 byte, con los 7 bits menos significativos indican la longitud de la trama PSDU mientras que el bit más significativo está reservado [7].
- **Unidad de datos de servicio PHY (PHY Service Data Unit):** este campo está constituido por la trama que se envía desde la subcapa MAC y tiene un tamaño máximo de 127 bytes [11] [12].

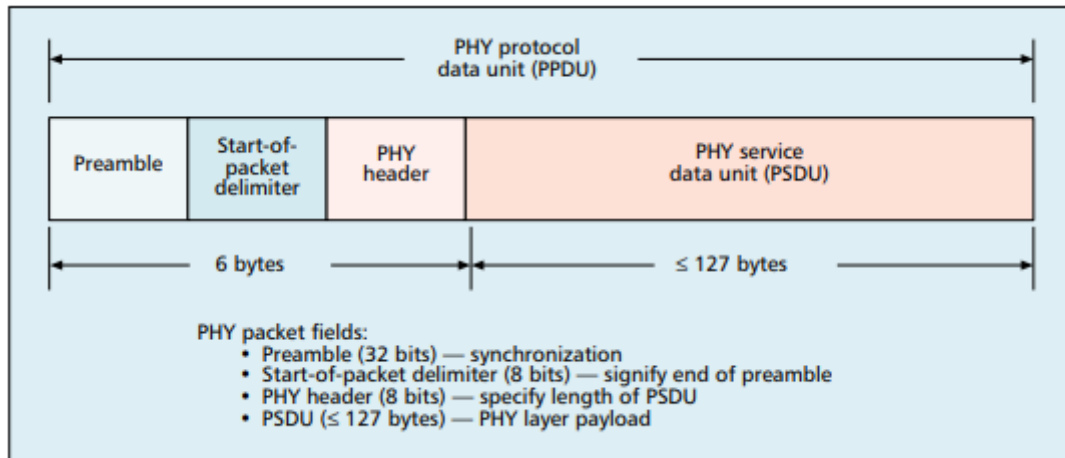


Figura 1.8. Formato de la PPDU [15].

❖ *Sensibilidad y potencia*

El estándar IEEE 802.15.4 en sus especificaciones menciona que en cuanto a la sensibilidad se tiene un valor de -85 dBm para una frecuencia de 2.4 GHz y un valor de -92 dBm para las frecuencias de 868 y 915 MHz, en cuanto a la potencia dependiendo de la aplicación este valor puede cambiar [6], de acuerdo con el presente trabajo realizado se pudo observar que la potencia máxima es de 3.5 dBm, pero se trabajó con la potencia por defecto que es de 3 dBm.

1.3.2.4.2 *Subcapa MAC (Medium Access Control)*

La subcapa de control de acceso al medio (MAC) IEEE 802.15.4 controla el acceso al canal de radio empleando el mecanismo CSMA/CA, además tiene las siguientes funciones [16]:

- Control de flujo a través de la entrega de tramas reconocidas.
- Validación de tramas.
- Mantener la sincronización de la red.
- Controlar la asociación.
- Generar beacons que permiten a los dispositivos encontrar una red existente
- Planificar el mecanismo de intervalo de tiempo garantizado.

❖ Estructura de las tramas MAC

Las estructuras de trama se han diseñado para mantener la complejidad al mínimo y, al mismo tiempo, hacerlas lo suficientemente robustas para la transmisión en un canal ruidoso [16]. El estándar 802.15.4 define cuatro estructuras de trama:

- **Trama de datos**

Se utiliza para la transferencia de datos [14].

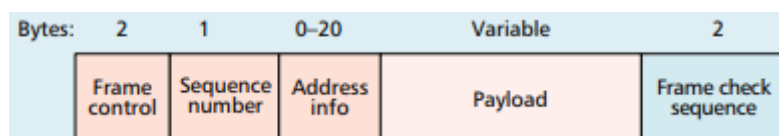


Figura 1.9. Formato de la trama de datos [15].

Como se observa en la Figura 1.9 la trama de datos está compuesta por los siguientes campos [6] [7] [10] [14]:

- ✓ *Control de trama (Frame Control)*: tiene un tamaño de 2 bytes e indica el tipo de trama que se va a enviar.
- ✓ *Número de secuencia (Sequence Number)*: posee un tamaño de 1 byte y permite reensamblar paquetes recibidos fuera de secuencia.
- ✓ *Dirección (Address)*: puede tener un tamaño entre 0 y 20 bytes, permite indicar las direcciones de origen y destino, así como también el identificador único de la PAN origen y destino.
- ✓ *Carga útil (Payload)*: es de tamaño variable y contiene la información (datos) que se va a transmitir.
- ✓ *Secuencia de chequeo de trama (Frame Check Sequence)*: posee un tamaño de 2 bytes y permite verificar que la trama recibida no tenga errores.

- **Trama beacon**

Esta trama es enviada por el coordinador de la PAN con el fin de sincronizar los dispositivos que pertenecen a la red [11].

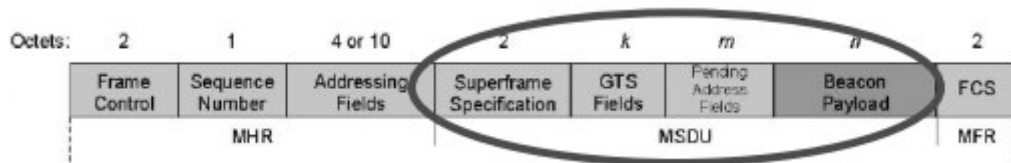


Figura 1.10. Formato de la trama beacon [14].

Como se observa en la Figura 1.10 la trama beacon posee campos que se explicaron en la trama de datos, por lo que a continuación se detallarán los nuevos campos que posee esta trama [6] [14]:

- ✓ *Especificación de supertrama (Superframe Specification):* proporciona información para que los dispositivos que pertenecen a la red sepan cuando pueden transmitir.
- ✓ *Campos GTS⁶ (GTS Fields):* contiene información correspondiente a los intervalos de tiempo garantizados.
- ✓ *Campos de dirección pendientes (Pending Address Fields):* contiene las direcciones de todos los dispositivos que tienen datos pendientes en el coordinador.
- ✓ *Carga útil Beacon (Beacon Payload):* lleva información que proviene de capas superiores.

- **Trama ACK⁷**

Esta trama es usada para confirmar cuando una trama se recibió correctamente [14], como se observa en la Figura 1.11 no posee campos diferentes a los que ya se explicaron en la trama de datos.

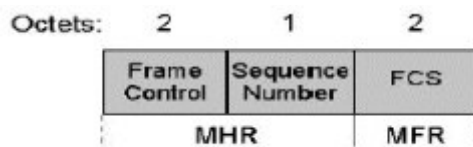


Figura 1.11. Formato de la trama ACK [14].

⁶ GTS por sus siglas en inglés “Guaranteed Time Slot” o Intervalo de tiempo garantizado es un período de tiempo en el cual un dispositivo puede transmitir sin usar CSMA/CA.

⁷ ACK por sus siglas en inglés “Acknowledgement” o acuse de recibo es una trama que el receptor envía al emisor para confirmar la recepción de un mensaje.

- **Trama de comandos MAC**

Esta trama permite ejecutar ciertas funciones, por ejemplo, la asociación o desasociación de dispositivos en la red, en sí se usa para configurar los dispositivos de la red.

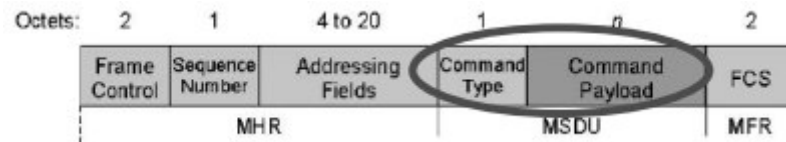


Figura 1.12. Formato de la trama de comandos MAC [14].

Como se observa en la Figura 1.12 la trama de comandos MAC posee campos que se explicaron en la trama de datos, por lo que a continuación se detallarán los nuevos campos que posee esta trama:

- ✓ *Tipo de comando (Command Type):* posee un tamaño de 1 byte. Existen un total de 9 tipos de comandos, este campo determina cual de dichos comandos se ejecutará [17].
- ✓ *Carga útil del comando (Command Payload):* en este campo se coloca el comando en sí y todos los parámetros asociados [17].

❖ *Modos de operación*

- **Modo Beacon Habilitado (Beacon Enable)**

Este modo es seleccionado por el coordinador PAN, utiliza una estructura de supertrama para realizar la comunicación entre los dispositivos de la red, dicho formato de supertrama es definido por el mismo coordinador PAN, el cual es enviado periódicamente a través de una trama beacon a los dispositivos que conforman la red [6].

- *Estructura de la supertrama*

El estándar LR-WPAN permite el uso opcional de una estructura de supertrama para aplicaciones que requieren ancho de banda dedicado para garantizar la latencia de comunicación [16].

La supertrama se compone de 16 intervalos de tiempo de igual tamaño agrupados en dos secciones: el período de acceso a la contención (CAP – Contention Access Period) y el período libre de contención (CFP –

Contention Free Period). El acceso al canal en los intervalos de tiempo se basa en la contienda, sin embargo, el coordinador de PAN puede asignar intervalos de tiempo a un solo dispositivo que requiera un ancho de banda dedicado o transmisiones de baja latencia. Estos intervalos de tiempo asignados se denominan intervalos de tiempo garantizados (GTS) y juntos forman un período libre de contención ubicado inmediatamente antes de la siguiente beacon, como se observa en la Figura 1.13. El tamaño del período sin contención puede variar según la demanda de los dispositivos de red asociados, cuando se emplean GTS, todos los dispositivos deben completar sus transacciones basadas en contención antes de que comience el período sin contención. El comienzo del período sin contención y la duración de la supertrama se comunican a los dispositivos de red conectados por el coordinador PAN en su baliza [15] [16].

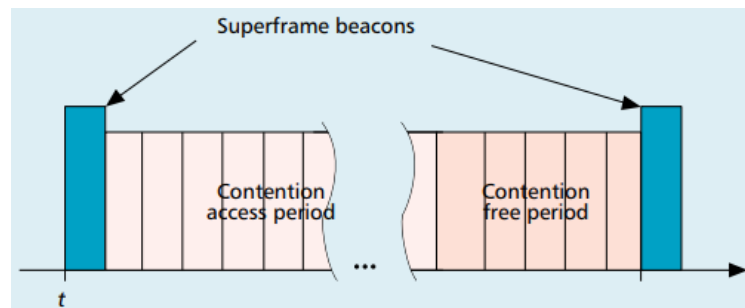


Figura 1.13. Estructura de la supertrama [15].

- **Modo Beacon Deshabilitado (Non Beacon Enable)**

Este modo no utiliza una estructura de supertrama para la comunicación por lo que los dispositivos que componen la red pueden enviar sus datos haciendo uso del mecanismo de acceso al medio CSMA/CA no ranurado [6].

- ❖ *Mecanismo de acceso al medio CSMA/CA [11] [12]*

Para ambos casos, se emplea el algoritmo de backoff⁸. En una red con el modo beacon habilitado el coordinador de la red PAN determina el periodo de backoff a través de su reloj interno, mientras que en una red con el modo beacon deshabilitado el reloj interno de cada dispositivo determina el periodo de backoff.

- **CSMA/CA ranurado (slotted CSMA/CA)**

⁸ Backoff: tiempo que un dispositivo debe esperar antes de intentar retransmitir una trama.

Este mecanismo se emplea cuando el modo beacon está habilitado. En este mecanismo los límites de los slots de la supertrama se alinean con los límites de los periodos de backoff de todos los dispositivos de la red PAN, cuando el coordinador inicia la trama beacon este se alinea con el inicio del primer periodo de backoff.

Cuando se va a transmitir una trama dentro de una red con el modo beacon habilitado, el dispositivo debe ejecutar el algoritmo de backoff antes de transmitir en un slot del CAP de la supertrama, si el canal no se encuentra ocupado inicia la trama en el siguiente slot disponible y si se encuentra ocupado ejecuta el algoritmo de backoff antes de intentar transmitir. En el periodo CFP no se utiliza el mecanismo CSMA/CA. También el beacon y las tramas ACK se transmiten sin usar CSMA/CA.

- **CSMA/CA no ranurado (unslotted CSMA/CA)**

Este mecanismo se emplea cuando el modo beacon está deshabilitado. Cuando un dispositivo desea transmitir a través de este mecanismo, en primer lugar, debe ejecutar el algoritmo de backoff, si el canal se encuentra disponible para transmitir información después que finaliza el periodo de backoff el dispositivo empieza a enviar datos, en caso de que el canal no se encuentre disponible, vuelve a ejecutar nuevamente el algoritmo de backoff.

❖ *Modelo de transferencia de datos [19]*

Existen tres tipos de transacciones de transferencia de datos:

- **Transferencia de datos a un coordinador en el que un dispositivo transmite los datos**

Este tipo de transferencia de datos se puede dar tanto en una red con el modo beacon habilitado como se observa en la Figura 1.14, así como también en una red con el modo beacon deshabilitado como se observa en la Figura 1.15. Además, se puede hacer uso de la trama ACK para confirmar la recepción de los datos.

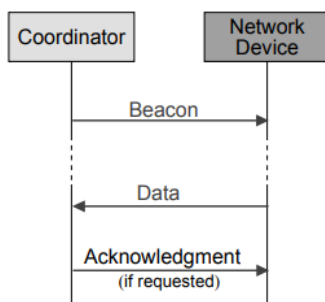


Figura 1.14. Comunicación con modo beacon habilitado [19].

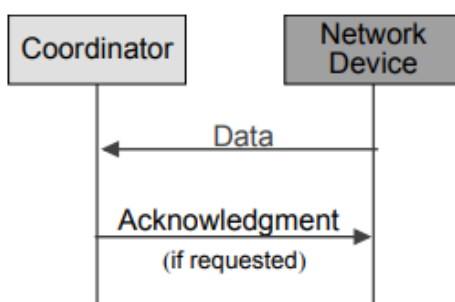


Figura 1.15. Comunicación con modo beacon deshabilitado [19].

- **Transferencia de datos de un coordinador en el que el dispositivo recibe los datos**

Este tipo de transferencia de datos se puede dar tanto en una red con el modo beacon habilitado como se observa en la Figura 1.16, así como también en una red con el modo beacon deshabilitado como se observa en la Figura 1.17. Además, al igual que en el caso anterior se puede hacer uso de la trama ACK para confirmar la recepción de los datos.

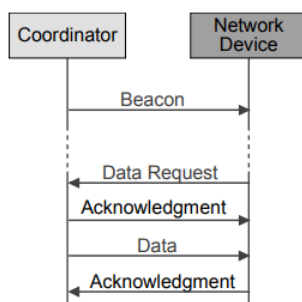


Figura 1.16. Comunicación con modo beacon habilitado [19].

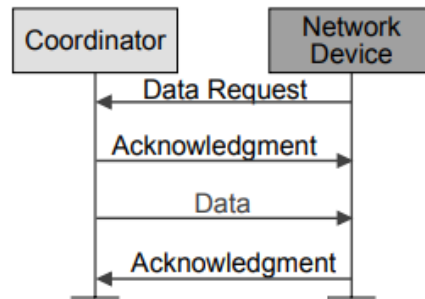


Figura 1.17. Comunicación con modo beacon deshabilitado [19].

- **Transferencia de datos entre dos dispositivos pares**

Cada dispositivo puede comunicarse con cualquier otro dispositivo en su rango de cobertura. Para hacer esto de manera efectiva, los dispositivos que deseen comunicarse deberán sincronizarse entre sí. En caso de usar CSMA/CA no ranurado es necesario tomar otras medidas para lograr la sincronización.

1.3.3 TRANSCEPTOR ATZB-256RFR2-XPRO

Como se observa en la Figura 1.18 el transceptor ATZB-256RFR2-XPRO es un kit que está conformado por una placa denominada Board Xplained PRO que es de mayor tamaño y se energiza mediante USB⁹ y por otra placa llamada ZigBit Extension que se energiza mediante dos pilas del tipo AAA, estas dos placas se pueden conectar en caso de ser necesario según la aplicación que se desee realizar. Para el presente trabajo se optó por hacer uso de la funcionalidad que permite a la placa ZigBit Extension trabajar como un nodo independiente operando con el estándar IEEE 802.15.4 como se observa en la parte derecha de la Figura 1.18. Las dos placas mencionadas trabajan con el microcontrolador ATmega256RFR2, por lo que de aquí en adelante al nodo inalámbrico que se configurará se lo denominará “transceptor ATZB-256RFR2-XPRO” en general.

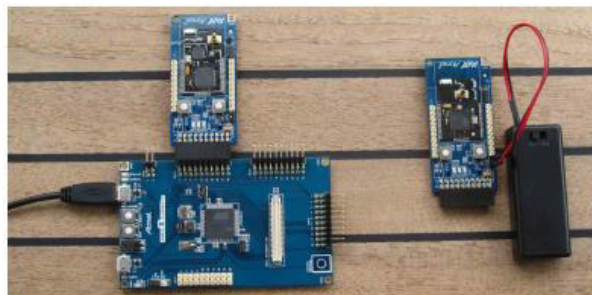


Figura 1.18. Transceptor ATZB-256RFR2-XPRO [20].

⁹ USB por sus siglas en inglés “Universal Serial Bus” es un bus de comunicaciones que sigue un estándar que define cables, conectores y protocolos usados en un bus para comunicar, conectar y proveer de alimentación eléctrica entre dispositivos electrónicos.

1.3.3.1 Características

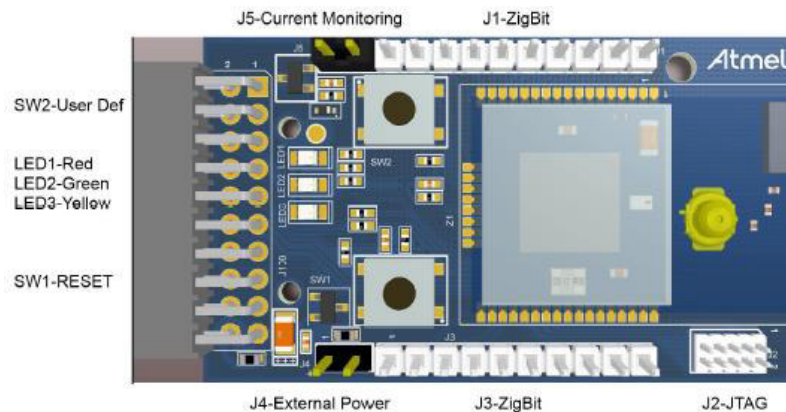


Figura 1.19. Descripción general del transceptor [20].

Como se observa en la Figura 1.19 el transceptor ATZB-256RFR2-XPRO posee las siguientes características [20]:

- **J1-ZigBit y J3-ZigBit:** cada uno consta de 10 pines y brindan acceso a todos los pines ZigBit no enrutados a la interfaz J100 Xplained PRO.
- **J2-JTAG:** corresponde a los pines de programación JTAG que suele utilizar JTAGICE.
- **J4-External Power:** permite la conexión de una fuente de alimentación externa cuando no está insertada en una placa Board Xplained PRO. El conector tiene protección contra voltaje inverso.
- **J5-Current Monitoring:** permite medir la corriente usando un multímetro externo, se debe conectar el multímetro entre J5-1 y J5-2. Es importante mencionar que cuando no se está midiendo la corriente, se debe insertar un puente entre J5-1 y J5-2 para proporcionar energía al transceptor.
- **LEDs:** existen tres LED disponibles (rojo, verde y amarillo) para uso del software de la aplicación. En la Figura 1.20 se observa el pin que debe ser configurado para cada uno de los LEDs al momento de realizar el código en Atmel Studio 7.0.

LED	ZigBit pin	256RFR2 pin
LED1 - Red	17	31 - PD6
LED2 - Green	18	16 - PG2
LED3 - Yellow	10	48 - PE2

Figura 1.20. Pines de los LEDs [20].

- **Button:** existen dos botones: uno RESET y otro para uso general que puede ser definido por el usuario. En la Figura 1.21 se observa el pin que debe ser configurado para el botón SW2 al momento de realizar el código en Atmel Studio 7.0.

Button	Function	ZigBit pin	256RFR2 pin
SW1	ZigBit Reset, press to reset	5	12 - RSTN
SW2	User defined high signal, press to negate	22	46 - PE0

Figura 1.21. Pines de los botones [20].

1.3.3.2 Programación

Para grabar el código realizado en Atmel Studio 7.0 en el transceptor ATZB-256RFR2-XPRO se deben seguir los siguientes pasos [20]:

- 1) Conecte el USB JTAGICE a la PC.
- 2) Conecte el JTAGICE al conector J2-JTAG de la Figura 1.19.
- 3) Vaya a Atmel Studio: Herramientas → Programación de dispositivos, seleccione el JTAGICE conectado como Herramienta y haga clic en Aplicar.
- 4) Seleccione el dispositivo ATmega256RFR2.
- 5) Seleccione "Memorias" y localice el archivo hexadecimal o elf de origen y haga clic en Programa.

En la Figura 1.22 se observa al dispositivo ATMEL-ICE para grabar el código en los transceptores al cual se hace referencia como JTAGICE en los pasos mencionados anteriormente.



Figura 1.22. Programador ATMEL-ICE [20].

1.3.3.3 Microcontrolador ATmega256RFR2

El ATmega256RFR2 es un microcontrolador CMOS¹⁰ de 8 bits de bajo consumo basado en la arquitectura RISC¹¹ combinada con un transceptor de alta velocidad de datos para la banda ISM de 2,4 GHz. Al ejecutar poderosas instrucciones en un solo ciclo de reloj, el dispositivo logra rendimientos cercanos a 1 MIPS¹² por MHz, lo que permite al diseñador del sistema optimizar el consumo de energía en comparación con la velocidad de procesamiento.

El transceptor de radio proporciona altas velocidades de datos desde 250 kbps hasta 2 Mbps, manejo de tramas, excelente sensibilidad del receptor y alta potencia de salida de transmisión que permite una comunicación inalámbrica muy robusta [21].

1.3.4 ENTORNO DE DESARROLLO ATMEL

Atmel Studio 7.0 es el entorno de desarrollo integrado (IDE¹³) para desarrollar y depurar aplicaciones de microcontrolador (MCU) basadas en SMART ARM¹⁴ y AVR¹⁵. Studio 7 IDP le brinda un entorno fluido y fácil de usar para escribir, compilar y depurar sus

¹⁰ CMOS por sus siglas en inglés “Complementary Metal-Oxide-Semiconductor” es una tecnología de fabricación de circuitos integrados que está diseñado para un bajo consumo de energía.

¹¹ RISC por sus siglas en inglés “Reduce Instruction Set Computer” es una arquitectura utilizada en microcontroladores para incrementar la velocidad de procesamiento gracias al número definido de instrucciones que soportan.

¹² MIPS por sus siglas en inglés “Millions of Instructions per Second” es una forma de medir la cantidad de instrucciones que un microcontrolador ejecuta por segundo.

¹³ IDE por sus siglas en inglés “Integrated Development Environment” es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

¹⁴ ARM por sus siglas en inglés “Advanced RISC Machine” es una arquitectura diseñada con el fin de lograr un procesamiento simplificado y que consuma menos energía.

¹⁵ AVR hace referencia a la familia de microcontroladores RISC del fabricante estadounidense Atmel.

aplicaciones escritas en C/C++ o código ensamblador. También se conecta a la perfección a los depuradores y kits de desarrollo de Atmel [23].

1.3.4.1 Arquitectura

El software principal de la pila MAC consta de tres capas, además una aplicación que se ha desarrollado basándose en el estándar IEEE 802.15.4 hace uso de dichas capas, las cuales se indican a continuación [22]:

- Capa de abstracción de plataforma – PAL (Platform Abstraction Layer)
- Capa de abstracción del transceptor – TAL (Transceiver Abstraction Layer)
- Capa MAC Core – MCL (MAC Core Layer)

1.3.4.1.1 Capa PAL

Esta capa proporciona una interfaz para los siguientes componentes [22]:

- Temporizador base para ejecutar el temporizador de software.
- Interrupción GPIO¹⁶ para acceso de transceptor externo.
- Acceso SPI¹⁷ para acceso de transceptor externo.
- Acceso a almacenamiento (Flash o NVM o EEPROM).

1.3.4.1.2 Capa TAL

Contiene la funcionalidad específica del transceptor utilizada para la compatibilidad con el estándar IEEE 802.15.4 y proporciona interfaces a la capa principal de MAC, que es independiente del transceptor subyacente. Los siguientes componentes se implementan dentro de la capa TAL [22]:

- Unidad de transmisión de tramas.
- Unidad de recepción de tramas.
- Máquina de estados.
- Almacenamiento de PIB TAL (TAL PAN Information Base).

¹⁶ GPIO por sus siglas en inglés “General Purpose Input/Output” es un pin genérico en un chip, cuyo comportamiento se puede programar por el usuario.

¹⁷ SPI por sus siglas en inglés “Serial Peripheral Interface” es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

- Módulo CSMA.
- Escaneo de detección de energía.
- Administración de energía.
- Manejo de interrupciones.
- Inicialización y reinicio.

1.3.4.1.3 Capa MCL

Abstrae e implementa el comportamiento compatible con el estándar IEEE 802.15.4 para el soporte de red habilitado y no habilitado para tramas beacon. Los bloques de construcción implementados son [22]:

- Servicio de datos MAC.
- Servicio de administración de MAC (inicio, asociación, escaneo, sondeo, etc.).
- Administrador de beacons.
- Procesador de trama entrante MAC.

1.3.4.1.4 Funcionamiento de las capas PAL, TAL, MCL

Una aplicación puede utilizar cualquier capa que desee según la funcionalidad requerida. Una aplicación que se basa en un estándar IEEE 802.15.4 utiliza la MAC-API basada en la pila creada por PAL, TAL y MCL. MAC-API permite que la aplicación interactúe con la pila MAC.

2. METODOLOGÍA

Para el desarrollo del presente trabajo de titulación se utilizó investigación aplicada, implementando un algoritmo que permite la asignación de direcciones a los nodos sensores utilizando el estándar IEEE 802.15.4 y una vez realizado este proceso existirá comunicación de extremo a extremo. El algoritmo se comprobó en un prototipo que consta de transceptores ATZB-256RFR2-XPRO los cuales son programados mediante el uso del software Atmel Studio 7, de esta manera se pueden realizar las pruebas necesarias para posteriormente interpretar los resultados obtenidos. El algoritmo implementado consta de dos partes que tienen como objetivo:

- Asignar direcciones en los transceptores ATZB-256RFR2-XPRO esto se dará en un escenario con un ambiente ideal, es decir en el que no existen fallos, en un escenario con enlaces ruidosos en los cuales puede que el nodo no procese alguna trama, también en un escenario con nodos caídos en los que algún nodo deja de funcionar por lo que al solucionar el inconveniente sucedido deba ingresar nuevamente a la red.
- Enviar datos de extremo a extremo que se puede dar entre dos nodos de la red o entre un nodo de la red y el Gateway

2.1. DESCRIPCIÓN DEL ALGORITMO

2.1.1 ALGORITMO PARA LA ASIGNACIÓN DE DIRECCIONES

Para el proceso de asignación de direcciones se hará uso del direccionamiento corto que está contenido dentro de las características del estándar IEEE 802.15.4, por medio de este algoritmo se obtendrá un direccionamiento jerárquico que permitirá formar una topología tipo árbol, cabe aclarar que se implementó el algoritmo solamente haciendo uso de la información de capa enlace. En la Figura 2.1 se muestra la red de sensores inalámbricos previo a la realización del proceso de direccionamiento.

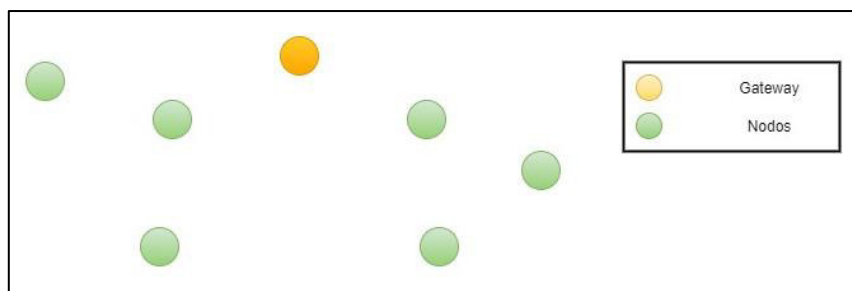


Figura 2.1. Topología previa al proceso de direccionamiento.

En el proceso de asignación de direcciones se tomará en cuenta a la topología tipo árbol por niveles, es decir, el primer nivel estará compuesto por el Gateway, el segundo nivel estará compuesto por los nodos a los que les asigne la dirección el Gateway, el tercer nivel estará compuesto por los nodos a los que les asigne la dirección los nodos a los cuales les asigno el Gateway y así sucesivamente continuarán aumentando los niveles. Para determinar el número de bits necesarios para la asignación de direcciones de acuerdo con el número de nodos se tomará en cuenta la información de la Tabla 2.1.

Tabla 2.1. Cantidad de bits usados para la dirección del coordinador [12].

Número de nodos	Número de bits necesarios	Número de bits en hexadecimal
1-3	2	0x02
4-7	3	0x03
8-15	4	0x04
16-31	5	0x05
32-63	6	0x06
64-127	7	0x07
128-255	8	0x08
256-511	9	0x09
512-1023	10	0x0A
1024-2047	11	0x0B
2048-4095	12	0x0C
4096-8191	13	0x0D

El estándar IEEE 802.15.4 determina 16 bits para el direccionamiento corto por lo cual se tendrá un bit asignado para la dirección del Gateway, así que quedarán 15 bits para las direcciones de los nodos pertenecientes al segundo nivel.

Tabla 2.2. Cantidad de bits usados para la dirección del coordinador.

Tipo de transceptor	Bit del Gateway	Bits disponibles para asignar direcciones (15)	Dirección hexadecimal
Gateway	0	000000000000000	0x0000

Cuando el Gateway recibe respuesta de los nodos que se encuentran en su zona de cobertura solicitando que se les asigne una dirección, este realiza el cálculo de los bits necesarios para la asignación de acuerdo con el número de nodos de los cuales recibió dicha respuesta. Por ejemplo, como se observa en la sección de “ALCANCE” en el capítulo 1 en la Figura 1.1, se necesita asignar la dirección a dos nodos por lo que se necesitarán 2 bits que permiten direccionar hasta tres nodos de acuerdo con la Tabla 2.1 que nos proporcionan las siguientes combinaciones (00, 01, 10, 11), en este caso la primera de las combinaciones no se utilizará ya que corresponde a la dirección usada por el Gateway. Como se observa en la Figura 2.2 los nodos 1 y 2 pertenecen al segundo nivel, además en la Tabla 2.3 se observa los bits utilizados para el direccionamiento de dichos nodos.

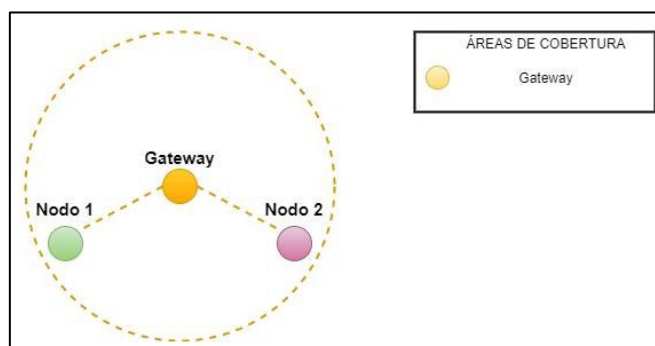


Figura 2.2. Topología tipo árbol con 2 niveles.

Tabla 2.3. Cantidad de bits usados para direccionar los nodos del segundo nivel [12].

Tipo de transceptor	Bit del Gateway	Bits usados para el segundo nivel (2)	Bits disponibles para asignar direcciones (13)	Dirección hexadecimal
Gateway	0	00	0000000000000	0x0000
Nodo 1	0	01	0000000000000	0x2000
Nodo 2	0	10	0000000000000	0x4000

-----	0	11	000000000000	0x6000
-------	---	----	--------------	--------

Una vez que los nodos tienen una dirección asignada se convierten en nodos asignadores, por lo que son capaces de asignar direcciones a los nodos que se encuentren dentro de su zona de cobertura. Cuando el nodo asignador recibe respuesta de los nodos que se encuentran en su zona de cobertura solicitando que se les asigne una dirección, este realiza el cálculo de los bits necesarios para la asignación de acuerdo con el número de nodos de los cuales recibió dicha respuesta. Por ejemplo, como se observa en la sección de “ALCANCE” en el capítulo 1 en la Figura 1.1, se necesita asignar la dirección a dos nodos por lo que se necesitarán 2 bits que permiten direccionar hasta tres nodos de acuerdo con la Tabla 2.1 que nos proporcionan las siguientes combinaciones (00, 01, 10, 11), esto se utilizará tanto para el caso del Nodo 1 como para el caso del nodo 2, además en este caso la primera de las combinaciones no se utilizará ya que corresponde a la dirección usada por el Gateway. Como se observa en la Figura 2.3 los nodos 1 y 2 pertenecen al segundo nivel y los nodos 1.1, 1.2, 2.1, 2.2 pertenecen al tercer nivel, además en la Tabla 2.4 se observa los bits utilizados para el direccionamiento de los nodos dentro de la zona de cobertura del Nodo 1 y en la Tabla 2.5 se observa los bits utilizados para el direccionamiento de los nodos dentro de la zona de cobertura del Nodo 2.

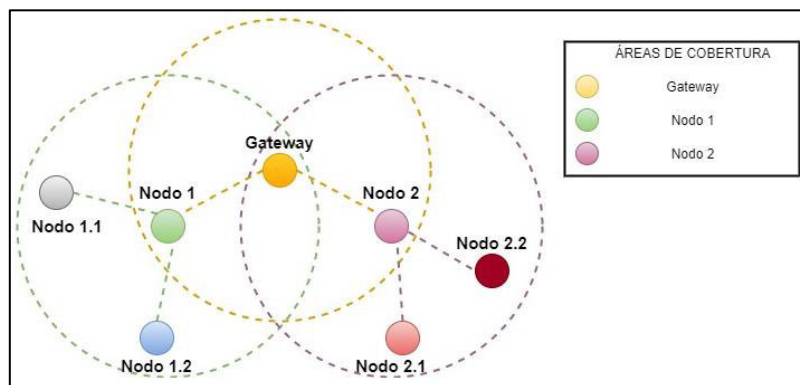


Figura 2.3. Topología tipo árbol con 3 niveles.

Tabla 2.4. Cantidad de bits usados para direccionar los nodos del tercer nivel (Nodo 1) [12].

Tipo de transceptor	Bit del Gateway	Bits usados para el segundo nivel (2)	Bits usados para el tercer nivel (2)	Bits disponibles para asignar direcciones (11)	Dirección hexadecimal
Nodo 1	0	01	00	00000000000	0x2000
Nodo 1.1	0	01	01	00000000000	0x2800
Nodo 1.2	0	01	10	00000000000	0x3000
-----	0	01	11	00000000000	0x3800

Tabla 2.5. Cantidad de bits usados para direccionar los nodos del tercer nivel (Nodo 2) [12].

Tipo de transceptor	Bit del Gateway	Bits usados para el segundo nivel (2)	Bits usados para el tercer nivel (2)	Bits disponibles para asignar direcciones (11)	Dirección hexadecimal
Nodo 2	0	10	00	00000000000	0x4000
Nodo 2.1	0	10	01	00000000000	0x4800
Nodo 2.2	0	10	10	00000000000	0x5000
-----	0	10	11	00000000000	0x5800

2.1.1.1 Algoritmo de asignación de direcciones por orden de llegada de oferta de asignación

Una vez que se energizan los transceptores empiezan como “nodos”, es decir todos tienen las mismas funciones al iniciar el proceso de direccionamiento, para conocer la dirección aleatoria provisional que poseen se envían a sí mismos una trama con el mensaje “MY_ID”, esto permitirá identificar quién le asigno la dirección fija además permitirá observar que dicha dirección aleatoria se ha cambiado por la dirección fija.

En el proceso de asignación de direcciones se tomará en cuenta la presencia de enlaces ruidosos y nodos caídos, para la simulación de estos escenarios se debe presionar una vez el botón SW2 que poseen los nodos. Para el caso de enlaces ruidosos se simuló que una vez enviado el broadcast por parte del nodo asignador el nodo no procese la trama, al contrario, una vez iniciado el proceso de direccionamiento espera un segundo y envía una trama unicast con el mensaje "FALLO" hacia el nodo asignador con el objetivo de que el nodo asignador le vuelva a enviar la trama unicast con la oferta de asignación de dirección y así continuar con el proceso normalmente.

Para el caso de nodos caídos se debe energizar el nodo y presionar el botón SW2, una vez hecho esto el nodo envía una trama en broadcast con el mensaje "CAIDO" con el fin de que alguno de los nodos asignadores que se encuentren dentro de su zona de cobertura le envíen una trama unicast con la oferta de asignación de dirección y así continuar con el proceso normalmente.

Para que el proceso de asignación de direcciones inicie se debe presionar dos veces el botón SW2 en el nodo que esté previsto sea el Gateway, hecho esto el nodo se convertirá en Gateway y deberá verificar si es un enlace arreglado, se implementó esta variable para hacer referencia al hecho de que haya recibido alguna trama por parte de los nodos que se encuentren en su zona de cobertura en el caso de que en dichos nodos se haya producido el escenario de enlace ruidoso o de nodo caído, el enlace será arreglado en el escenario que se presentan enlaces ruidosos o nodos caídos, mientras que el enlace no será arreglado en el escenario de un ambiente ideal. Cabe aclarar que el Gateway es el primer nodo asignador.

Una vez que se comprueba el estado del enlace se pueden dar dos casos:

- Si el enlace no es arreglado, es decir, el escenario es un ambiente sin fallos, en este caso el nodo asignador envía una trama en broadcast con el mensaje "L".
- Si el enlace es arreglado, es decir, el nodo asignador recibe tramas con los mensajes "FALLO" o "CAIDO", envía una trama unicast con el mensaje "L" a los nodos de los cuales recibió los mensajes "FALLO" o "CAIDO".

Una vez que el nodo asignador envía las tramas respectivas según el caso que se presente los nodos que se encuentren en la zona de cobertura recibirán la trama con el mensaje "L", luego aumentarán el contador de nodos asignadores de los cuales recibieron el mensaje ya que puede ser que reciba de uno o más nodos asignadores, además almacena las direcciones de dichos nodos asignadores e inicia el timer1.

Dentro del timer1 los nodos que recibieron el mensaje “L” comprueban el valor del contador de nodos asignadores, en este punto se pueden presentar dos casos:

- Si el contador de nodos asignadores es igual a uno, el nodo envía una trama de respuesta al nodo asignador con el mensaje “NEED_ID_L”.
- Si el contador de nodos asignadores es diferente de uno, el nodo envía una trama de respuesta con el mensaje “NEED_ID_L” al nodo asignador cuya dirección se encuentra en el primer lugar del arreglo de direcciones almacenadas.

Después de esto el nodo asignador recibe la trama de respuesta con el mensaje “NEED_ID_L”, luego aumenta el contador de nodos de los cuales recibió respuesta y almacena las direcciones de dichos nodos e inicia el timer2. Dentro del timer2 el nodo asignador calcula los bits disponibles para asignar direcciones de acuerdo con el número de nodos de los cuales recibió respuesta y a dichos nodos les envía una trama con la información para definir la dirección fija. Finalmente activa la etapa de comunicación en modo recepción.

Una vez que el nodo asignador realiza dicho proceso el o los nodos reciben la trama con la información para calcular la dirección fija y la definen, luego de esto pasan de ser nodos a nodos asignadores y continúan el proceso en el punto A como se indica en la Figura 2.4

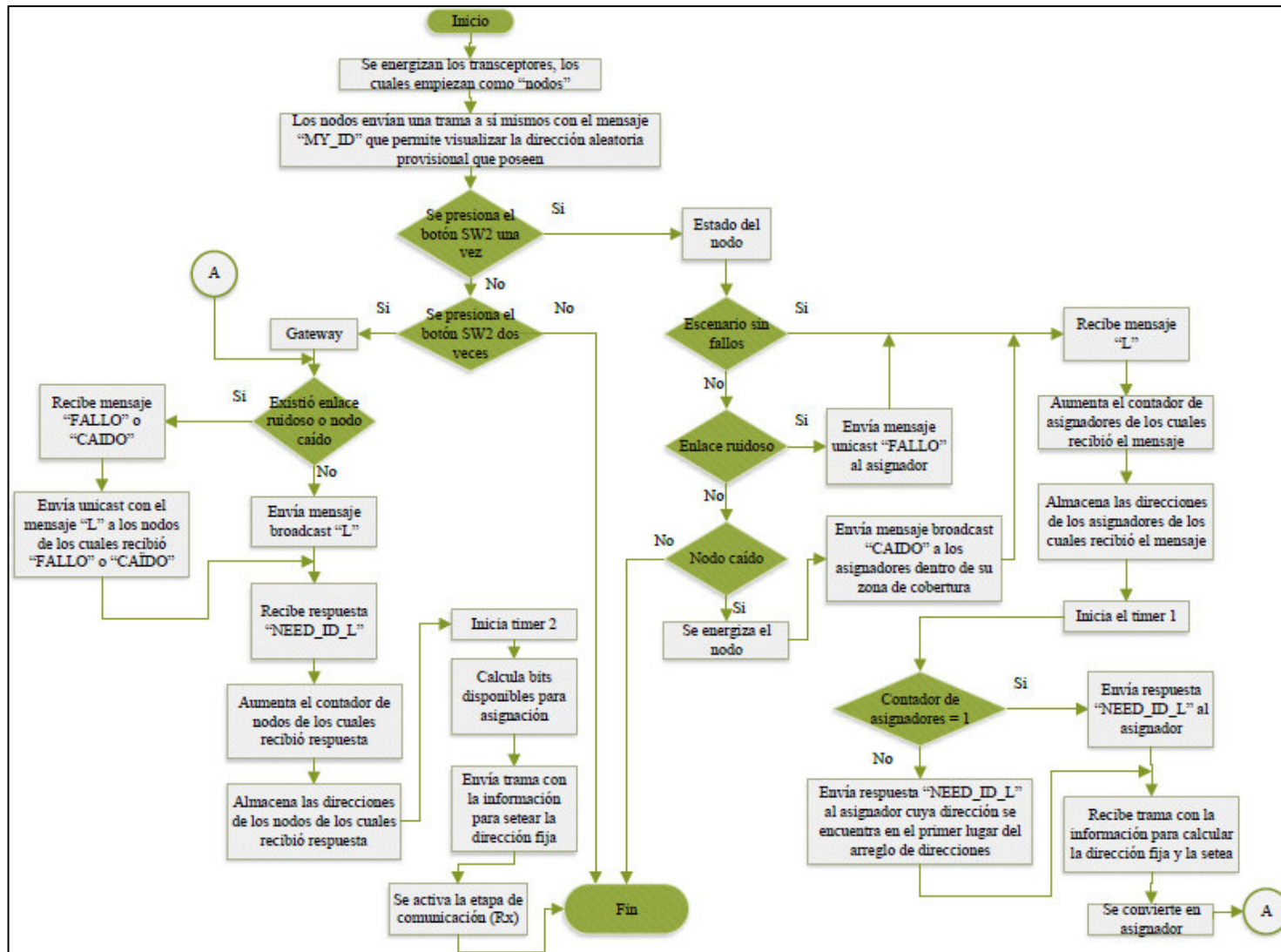


Figura 2.4. Diagrama de flujo del algoritmo de asignación de direcciones por orden de llegada de la oferta de asignación.

2.1.1.2 Algoritmo de asignación de direcciones por nivel de potencia de la trama recibida

La descripción del algoritmo de la sección 2.1.1.1 es similar para este algoritmo hasta el momento en que el nodo asignador comprueba el estado del enlace, por lo que a continuación se procederá a explicar cuáles son los cambios de este algoritmo en relación con el de la sección mencionada.

Una vez que se comprueba el estado del enlace se pueden dar dos casos:

- Si el enlace no es arreglado, es decir, el escenario es un ambiente sin fallos, en este caso el nodo asignador envía una trama en broadcast con el mensaje "P".
- Si el enlace es arreglado, es decir, el nodo asignador recibe tramas con los mensajes "FALLO" o "CAIDO", envía una trama unicast con el mensaje "P" a los nodos de los cuales recibió los mensajes "FALLO" o "CAIDO".

Una vez que el nodo asignador envía las tramas respectivas según el caso que se presente los nodos que se encuentren en la zona de cobertura recibirán la trama con el mensaje "P", luego aumentarán el contador de nodos asignadores de los cuales recibieron el mensaje ya que puede ser que reciba de uno o más nodos asignadores y también se extrae la potencia de la o las tramas recibidas, además almacena las direcciones de dichos nodos asignadores para luego iniciar el timer1.

Dentro del timer1 los nodos que recibieron el mensaje "P" comprueban el valor del contador de nodos asignadores, en este punto se pueden presentar dos casos:

- Si el contador de nodos asignadores es igual a uno, el nodo envía una trama de respuesta al nodo asignador con el mensaje "NEED_ID_P".
- Si el contador de nodos asignadores es diferente de uno, se ordena de mayor a menor las potencias de las tramas recibidas y el nodo envía una trama de respuesta con el mensaje "NEED_ID_P" al nodo asignador del cual se recibió la trama con mayor potencia.

Después de esto el nodo asignador recibe la trama de respuesta con el mensaje "NEED_ID_P", luego aumenta el contador de nodos de los cuales recibió respuesta y almacena las direcciones de dichos nodos e inicia el timer2. Dentro del timer2 el nodo asignador calcula los bits disponibles para asignar direcciones de acuerdo con el número de nodos de los cuales recibió respuesta y a dichos nodos les envía una trama con la

información para definir la dirección fija. Finalmente activa la etapa de comunicación en modo recepción.

Una vez que el nodo asignador realiza dicho proceso el o los nodos reciben la trama con la información para calcular la dirección fija y la definen, luego de esto pasan de ser nodos a nodos asignadores y continúan el proceso en el punto A como se indica en la Figura 2.5

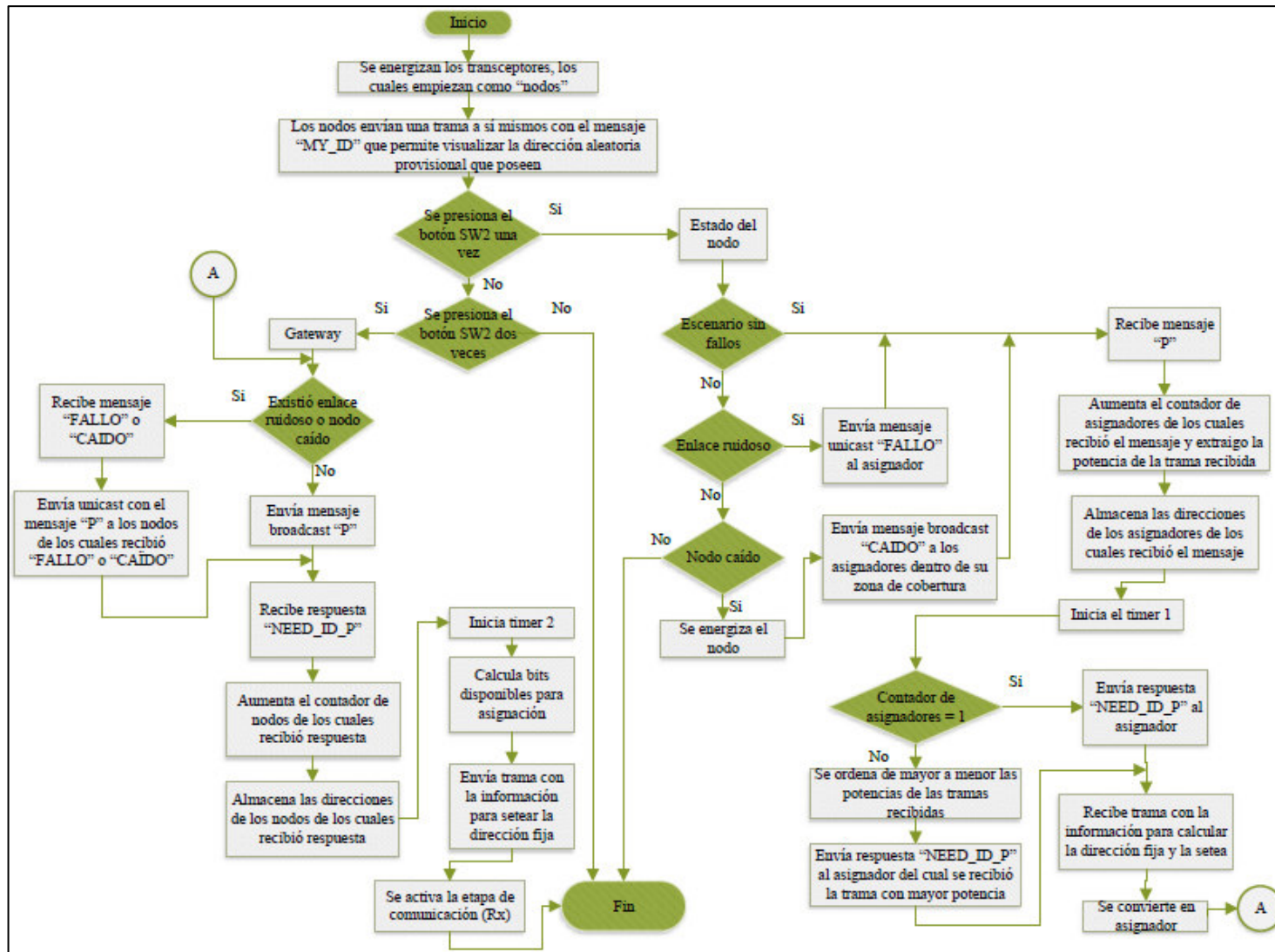


Figura 2.5. Diagrama de flujo del algoritmo de asignación de direcciones por nivel de potencia de la trama recibida.

2.1.1.3 Algoritmo de asignación de direcciones por número de saltos

Cabe aclarar que este algoritmo hace referencia al número de saltos para llegar al Gateway. En este algoritmo se añadió una función adicional al presionar el botón SW2 una vez, dicha función trata de simular un escenario en el que se pueda comprobar que realmente el nodo asignador es el que necesita un menor número de saltos para llegar Gateway, en este escenario al presionar el botón SW2 el nodo envía una trama en broadcast con el mensaje "SALTO" que será recibida por los nodos asignadores que se encuentren dentro de su zona de cobertura, una vez hecho esto el nodo asignador que necesite un menor número de saltos para llegar al Gateway le enviará un trama unicast al nodo con la oferta de asignación de dirección, este escenario es parecido al escenario de nodos caídos al cual se hace referencia en la sección 2.1.1.1.

La descripción del algoritmo de la sección 2.1.1.1 se utiliza como base para la creación del algoritmo de asignación de direcciones por número de saltos, por lo que a continuación se procederá a explicar cuáles son los cambios de este algoritmo en relación con el de la sección mencionada.

Una vez que se comprueba el estado del enlace se pueden dar dos casos:

- Si el enlace no es arreglado, es decir, el escenario es un ambiente sin fallos, en este caso el nodo asignador envía una trama en broadcast con el mensaje "S" acompañado del valor del contador de saltos, por ejemplo, el nodo asignador envía la trama con el mensaje "S2", esto quiere decir que el nodo que reciba esta trama extraerá el valor 2 y ese equivaldrá al número de saltos que dicho nodo tendría que dar para llegar al Gateway.
- Si el enlace es arreglado, es decir, el nodo asignador recibe tramas con los mensajes "FALLO", "CAIDO" o "SALTO", envía una trama unicast con el mensaje "S" acompañado del valor del contador de saltos a los nodos de los cuales recibió los mensajes "FALLO", "CAIDO" o "SALTO", tal como se explicó en el ejemplo del literal anterior.

Una vez que el nodo asignador envía las tramas respectivas según el caso que se presente aumenta el contador de saltos y los nodos que se encuentren en la zona de cobertura recibirán la trama con el mensaje "S" acompañado del valor del contador de saltos, luego aumentarán el contador de nodos asignadores de los cuales recibieron el mensaje ya que puede ser que reciba de uno o más nodos asignadores y también se

extrae el contador de saltos, además almacena las direcciones de dichos nodos asignadores para luego iniciar el timer1.

Dentro del tiempo definido por el timer1 los nodos que recibieron el mensaje "S" acompañado del valor del contador de saltos comprueban el valor del contador de nodos asignadores, en este punto se pueden presentar dos casos:

- Si el contador de nodos asignadores es igual a uno, el nodo envía una trama de respuesta al nodo asignador con el mensaje "NEED_ID_S".
- Si el contador de nodos asignadores es diferente de uno, se ordena de menor a mayor según el contador de saltos y el nodo envía una trama de respuesta con el mensaje "NEED_ID_S" al nodo asignador por el cual necesite menos saltos para llegar al Gateway.

Después de esto el nodo asignador recibe la trama de respuesta con el mensaje "NEED_ID_S", luego aumenta el contador de nodos de los cuales recibió respuesta y almacena las direcciones de dichos nodos e inicia el timer2. Dentro del timer2 el nodo asignador calcula los bits disponibles para asignar direcciones de acuerdo con el número de nodos de los cuales recibió respuesta y a dichos nodos les envía una trama con la información para definir la dirección fija. Finalmente activa la etapa de comunicación en modo recepción.

Una vez que el nodo asignador realiza dicho proceso, él o los nodos reciben la trama con la información para calcular la dirección fija y la definen, luego de esto pasan a ser nodos asignadores y continúan el proceso en el punto A como se indica en la Figura 2.6

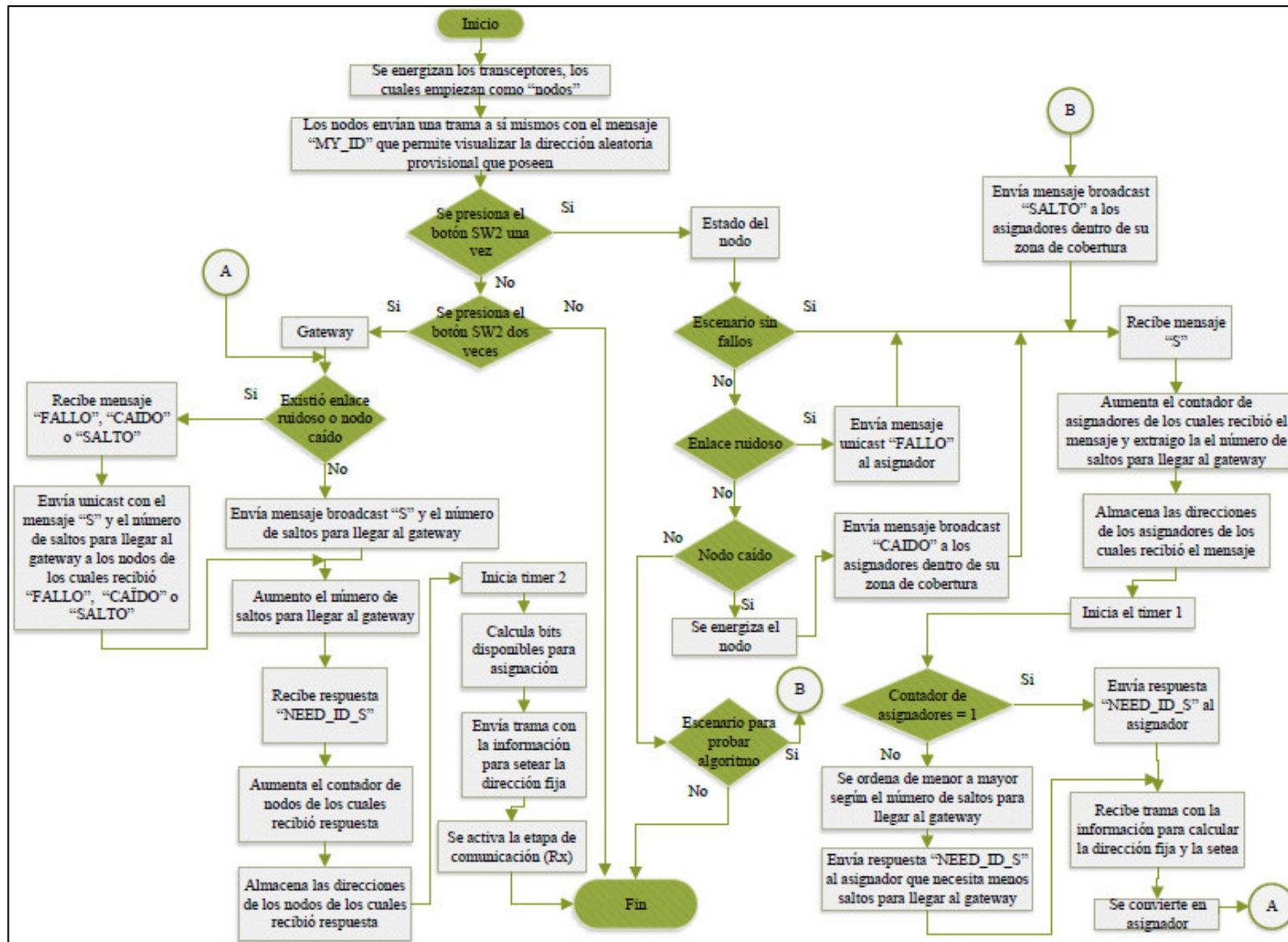


Figura 2.6. Diagrama de flujo del algoritmo de asignación de direcciones por número de saltos.

2.1.2 ALGORITMO PARA LA COMUNICACIÓN DE EXTREMO A EXTREMO

Para el proceso de comunicación de extremo a extremo cabe mencionar que esto implica, tanto la comunicación entre un nodo y el Gateway o entre dos nodos de la red. Dado que el Gateway básicamente es otro nodo de la red se utilizó el mismo diagrama de flujo para los dos casos que se mencionaron.

Una vez finalizada la etapa de asignación de direcciones, todos los nodos pertenecientes a la WSN tienen activada la etapa de comunicación en modo recepción, para activar la etapa de comunicación en modo transmisión para el nodo en el cual se configuró esta tarea se debe presionar el botón SW2 una vez, hecho esto se define la dirección de destino.

Cabe mencionar que el término “nodo actual” mencionado en la Figura 2.7, hace referencia al nodo que esté procesando el mensaje enviado en la etapa de comunicación que consta de la palabra “ENVIO” seguida de la dirección destino. Además, en la Figura 2.7 también se menciona el término “nodo rama”, el cual hace referencia a cada uno de los nodos a los cuales un nodo asignador les proporcionó la información para que calculen y definan su nueva dirección.

El nodo actual compara su dirección con la dirección destino, por lo que se pueden dar dos casos:

- Si las direcciones son iguales, no se envía ningún mensaje más, ya que dicho nodo es el destinatario.
- Si las direcciones son diferentes, el nodo actual envía a su nodo asignador un mensaje con la palabra “ENVIO” seguida de la dirección destino, cabe mencionar que el nodo asignador pasa a convertirse en nodo actual.

Cuando el nodo actual recibe el mensaje extrae los primeros cinco caracteres y realiza las siguientes comparaciones:

- Si los caracteres son iguales a la palabra “ENVIO” procede a extraer los cuatro caracteres restantes que hacen referencia a la dirección destino.
- Si los caracteres son diferentes a la palabra “ENVIO” la etapa de comunicación finaliza.

Al extraer los caracteres de la dirección destino, el nodo actual compara dicha dirección con la dirección o direcciones de sus nodos rama, por lo que se pueden presentar los siguientes casos:

- Si la dirección destino corresponde a la dirección de alguno de los nodos rama del nodo actual, se envía el mensaje con la palabra “ENVIO” seguida de la dirección destino a dicho nodo rama.
- Si la dirección destino no corresponde a la dirección de alguno de los nodos rama del nodo actual, se vuelve a realizar la primera comparación que se mencionó en esta sección.

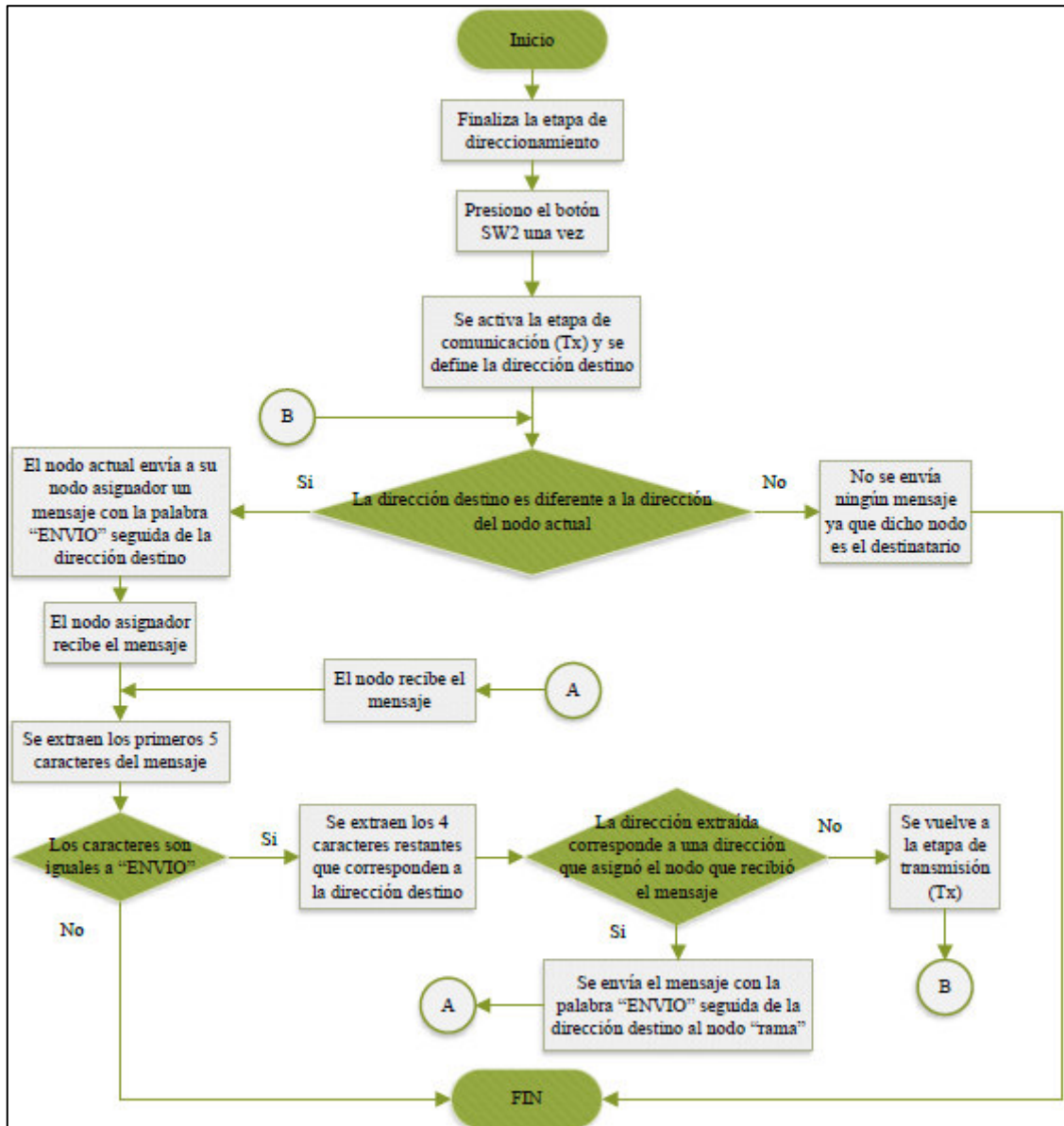


Figura 2.7. Diagrama de flujo del algoritmo para la comunicación de extremo a extremo.

2.2. CODIFICACIÓN DEL ALGORITMO

Tabla 2.6. Elementos necesarios para generar y comprobar el funcionamiento del código.

HARDWARE	
Herramienta	Cantidad
Laptop	1
Transceptor ATZB-256RFR2-XPRO	6
Transceptor RCB256RFR2	1
Atmel ICE (Quemador)	1
Sniffer CC2531 Dongle	1
Porta pilas AAA	6
SOFTWARE	
Sistema Operativo Windows 10	1
Atmel Studio 7.0	1
Wireless Composer for Atmel 7.0	1
Texas Instruments SmartRF Packet Sniffer	1

Para la codificación del algoritmo y la programación de los transceptores se utilizó los elementos mencionados en la Tabla 2.5, específicamente para la realización del código se hizo uso del software Atmel Studio 7.0, empleando el IEEE 802.15.4 MAC Project Wizard que es parte de la extensión Wireless Composer 7.0.

Una vez creado el proyecto se crean diferentes archivos de configuración que se muestran en el explorador de soluciones (Solution Explorer), como se observa en la Figura 2.7.

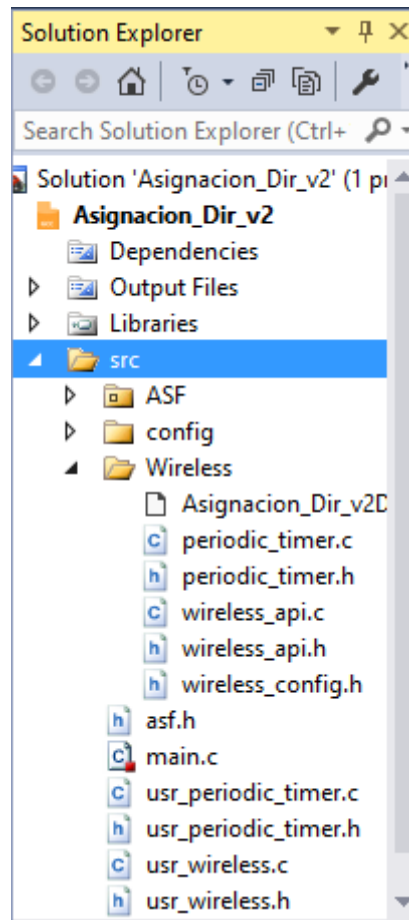


Figura 2.8. Solution Explorer con los archivos de configuración generados.

A continuación, se va a realizar una breve explicación de los archivos que se utilizaron para la implementación del código:

- **Archivo main.c:** inicializa las funciones básicas del transceptor.
- **Archivos de usuario:** como su nombre lo indica estos archivos deben ser modificados por el usuario para agregar la funcionalidad al programa, entre los archivos que se modificaron están: `usr_wireless.c` donde principalmente se creó el código, `usr_wireless.h` donde se crearon las estructuras necesarias para trabajar con las tramas, `usr_periodic_timer.c` y `usr_periodic_timer.h` donde se configura el tiempo que durarán los timers, así como las funciones de inicio y fin de estos.
- **Archivos Wireless:** estos archivos contienen las funciones que se usarán para configurar los transceptores, entre los archivos que se usaron están: `wireless_api.c` en el cual se definen las direcciones aleatorias, `wireless_api.h` donde se encuentran las firmas de las funciones para la transmisión de datos y `wireless_config.h` en el cual se encuentra la configuración inicial de los

transceptores que se realizó al crear el proyecto mediante el IEEE 802.15.4 MAC Project Wizard.

Para la realización del código en el archivo `usr_wireless.h` se definieron las siguientes estructuras.

```
#define maximo 101 // Constante para definir el valor máximo del payload de la trama IEEE 802.15.4
#define arreglo 10 // Constante para definir el valor máximo de los arreglos de direcciones, potencia y saltos.

typedef struct
{
    uint8_t longitud; // Longitud de trama.
    uint16_t FCF; // Control de trama.
    uint8_t nSec; // Número de secuencia de trama.
    uint16_t PAN_dest; // PAN de destino.
    uint16_t add_dest; // Dirección de destino.
    uint16_t add_origen; // Dirección de origen.
    char carga[maximo]; // Payload.
    uint16_t FCS; // Secuencia de verificación de trama.
} trama; // Nombre para declarar la estructura.

typedef struct
{
    uint16_t direccionCoord[arreglo]; // Arreglo de direcciones.
    int potencia[arreglo]; // Arreglo de potencias.
} seleccionPotencia; // Nombre para declarar la estructura.

typedef struct
{
    uint16_t direccionCoord[arreglo]; // Arreglo de direcciones.
    int saltos[arreglo]; // Arreglo de saltos.
} seleccionSaltos; // Nombre para declarar la estructura.
```

Código 2.1. Estructuras creadas en `usr_wireless.h`.

Para la realización del código en primer lugar en el archivo `usr_wireless.c` se definieron los siguientes elementos.

- **Declaración de librerías**

```
#include "usr_wireless.h" // Archivo de usuario que permite modificar y agregar lógica a la aplicación.
#include "wireless_config.h" // Archivo que permite la configuración inalámbrica básica.
#include "periodic_timer.h" // Archivo que permite configurar la duración, así como inicio y fin de los timers.
```

Código 2.2. Declaración de librerías.

- **Declaración de variables para la etapa de direccionamiento**

```
#define INICIO // Directiva para definir la porción de código en que se presiona el botón.
#define TXBROADCAST // Directiva para definir la transmisión de tramas.
#define RXBROADCASTCOORD // Directiva para definir la recepción y procesamiento de tramas enviadas por los asignadores.
#define RXRESPUESTANODO // Directiva para definir la recepción y procesamiento de tramas enviadas por los nodos.
#define RXNUEVADIR // Directiva para definir la recepción y procesamiento de tramas enviadas por los asignadores con la
// información para calcular la dirección fija.

#define MAX 10 // Constante para definir el valor máximo de los arreglos de direcciones y longitud de caracteres.
```

Código 2.3. Declaración de directivas y constantes.

```

// ENTERO SIN SIGNO DE 2 BYTES

uint16_t dirBroadcast = DST_ADDR; // Dirección para enviar mensajes tipo broadcast.
uint16_t dirAsignadores [MAX]; // Arreglo para que el nodo almacenen las direcciones de los asignadores que le ofrecen una dirección.
uint16_t dirNodos [MAX]; // Arreglo para que los asignadores almacenen las direcciones de los nodos que les responden.
uint16_t dirInicial = SRC_ADDR; // Dirección que se definirá al Gateway (0x0000).
uint16_t dirCalculada; // Dirección en la que se desplazan los bits según el número de bits disponibles para la asignación
// de direcciones.

uint16_t dirAsignada; // Dirección fija que será definida en el nodo que lo haya solicitado.
uint16_t temp = 0x31; // Variable que se le asigna el valor de 0x31 en hexadecimal que en texto equivale a 1.
uint16_t contador = 0x0001; // Variable que se le asigna el valor hexadecimal de 0x0001.
uint16_t numBits; // Variable para extraer de la trama el número de bits disponibles para la asignación de direcciones.
uint16_t dirEnlaceRuidoso; // Dirección a la que el nodo enviara la trama unicast en caso del escenario de enlace ruidoso.
uint16_t dirEnlaceArreglado; // Dirección a la que el asignador enviará la trama unicast en caso del escenario de enlace ruidoso,
// nodo caído o saltos.

uint16_t dirComparada; // Variable usada para almacenar las direcciones al momento de ordenar de mayor a menor las potencias.
// o de menor a mayor según el número de saltos.

// ENTERO SIN SIGNO DE 1 BYTE

uint8_t bitsUsados; // Variable que indica los bits que se usarán para la asignación de direcciones según el número de nodos.
uint8_t bitsDisponibles = 0x0F; // Variable que indica los bits disponibles para la asignación de direcciones, se inicializa con 0x0F en
// hexadecimal que equivale a 15 en decimal.

uint8_t phy_frame_len; // Variable para almacenar la longitud de la trama.

// ENTERO SIN SIGNO DE 1 BYTE

int8_t rssi_base_val; // Variable usada para almacenar el valor base del indicador de la fuerza de la señal recibida (RSSI).
int8_t ed_value; // Variable para almacenar el valor de la potencia en dBm.
int8_t potComparada; // Variable usada para almacenar el valor de la potencia al momento de ordenarlas de mayor a menor.

// VARIABLES ENTERAS

int cont = 0; // Contador que almacena la cantidad de veces que se presiona el botón.
int contCoords = 0; // Contador que aumenta el nodo según las tramas que reciba por parte de los asignadores ofreciendo darle
// una dirección.
int contNodos = 0; // Contador que aumentan los asignadores según las tramas que reciba por parte de los nodos solicitando
// una dirección.
int contEnvioBroadcast = 0; // Contador para determinar el número de tramas que envía el Gateway.
int contBroadcastAsignador = 0; // Contador para determinar el número de tramas que envían los nodos que se convierten en asignadores.
int contSaltos = 1; // Contador que indica el número de saltos que deberán los nodos para llegar al gateway.
int saltosComparacion; // Variable usada para almacenar el valor del número de saltos al momento de ordenarlos de menor a mayor.
int k=0; // Variable usada para incrementar las posiciones en los arreglos de direcciones.
int l=0; // Variable usada para incrementar las posiciones en los arreglos de direcciones.

```

Código 2.4. Declaración de variables enteras con y sin signo.

```

// VARIABLES BOOLEANAS

bool gateway = false; // Variable que define al trasceptor como Gateway.
bool direccionamiento = true; // Variable que indica el inicio de la etapa de direccionamiento.
bool dirAleatoriaNodo = true; // Variable que indica si el nodo tiene una dirección aleatoria provisional.
bool rxTramasAsignador = true; // Variable que permite a los nodos recibir las tramas enviadas por los asignadores.
bool transmitirDirs = false; // Variable que permite a los asignadores enviar tramas a los nodos con la información necesaria
// para calcular la dirección fija y definirla.

bool dirEnviada = false; // Variable que indica que el nodo respondió a la oferta realizada por el nodo asignador para la
// asignación de una dirección.

bool rxRespNodo = false; // Variable que permite a los asignadores recibir tramas enviadas por los nodos.
bool asignador = false; // Variable que indica que el nodo se convirtió en asignador.
bool asignacion = false; // Variable que permite al nodo calcular la dirección fija y definirla.
bool enlaceRuidoso = false; // Variable que permite simular el escenario de enlaces ruidosos.
bool transmitirEnlaceRuidoso = false; // Variable que permite al nodo enviar la trama unicast al asignador en el escenario de enlaces
// ruidosos.

bool ordenLlegada = false; // Variable que permite determinar que el criterio de selección de ruta será por orden de llegada
// de la oferta de asignación.

bool ordenPotencia = false; // Variable que permite determinar que el criterio de selección de ruta será por el nivel de potencia
// de la trama recibida.

bool ordenSaltos = false; // Variable que permite determinar que el criterio de selección de ruta será por número de saltos.
bool escenarioSaltos = false; // Variable que permite simular el escenario de saltos y comprobar que el nodo responde a la oferta
// del asignador por el que debe dar menos saltos para llegar al Gateway.

bool enlaceArreglado = false; // Variable que permite identificar si se produjo un escenario de enlaces ruidosos, nodos caídos o
// saltos.

bool nodoCaido = false; // Variable que permite simular el escenario de nodos caídos.

```

Código 2.5. Declaración de variables booleanas.

```

// VARIABLES TIPO CARACTER

char nuevaDir[MAX];           // Variable en la que se almacena la información para que el nodo setee su dirección fija.
char payload_rx[MAX];        // Variable en la que se almacena el payload de las tramas recibidas.
char payload_tx[MAX];        // Variable que almacena el payload que se enviará en las tramas.
char mensajeRx[MAX];         // Variable que permite extraer el payload de la trama recibida en caso de que se haya enviado "FALLO",
                             // "CAIDA", "SALTO", "ENVIO" o "CONFI".
char envioSaltos[MAX];       // Variable que almacena el valor del contador de saltos para enviarlo en la trama.

// ESTRUCTURAS

trama trama_rx;              // Variable que contiene la estructura de la trama creada en el archivo usr_wireless.h
seleccionPotencia ruta;      // Variable que contiene la estructura para almacenar la potencia y dirección de la trama recibida, dicha
                             // estructura fue creada en el archivo usr_wireless.h
seleccionSaltos rutaS;       // Variable que contiene la estructura para almacenar el número de saltos y dirección de la trama recibida,
                             // dicha estructura fue creada en el archivo usr_wireless.h

```

Código 2.6. Declaración de variables tipo char.

- **Declaración de variables para la etapa de comunicación**

```

// VARIABLES BOOLEANAS

bool comunicacion = false;    // Variable que permite activar la etapa de comunicación.
bool transmitComunicacion = false; // Variable que permite iniciar la transmisión en la etapa de comunicación.
bool transmitRespuesta = false; // Variable que permite enviar un mensaje de confirmación en caso que la trama haya llegado a su destino.
bool transmitRama = false;    // Variable que permite enviar una trama a un nodo que es rama de un asignador.
bool esRama = false;          // Variable que permite conocer si un nodo es una rama de un nodo asignador.

// VARIABLES TIPO CARACTER

char mensajeCom[MAX];         // Variable que almacena el mensaje extraído del payload de la trama recibida.
char mensajeDestino[MAX];     // Variable que almacena el mensaje que se enviará al iniciar la comunicación.
char dirDestinoConvertida [MAX] ; // Variable que almacena la dirección destino convertida de formato hexadecimal a char.
char dirPayload [MAX];        // Variable que almacena la dirección destino extraída del payload de la trama recibida
                             // para convertirla a formato hexadecimal.

```

Código 2.7. Declaración de variables booleanas y tipo char.

```

// ENTERO SIN SIGNO DE 2 BYTES

uint16_t dirRecibida;         // Dirección en la que se almacena la dirección destino que se convirtió de char a hexadecimal.
uint16_t dirDestino;          // Dirección que se define al presionar el botón e indica el destino de la trama enviada.
uint16_t dirComunicacion[MAX]; // Arreglo de direcciones que permite a los asignadores conocer a quien le proporcionaron una
                             // dirección.
uint16_t dirRama;             // Dirección a la cual se debe reenviar la trama recibida en caso de que el nodo sea rama de algún
                             // asignador.
uint16_t dirCalculadaCom;     // Dirección que se obtiene una vez que se desplazan los bits según la cantidad de bits disponibles
                             // para la asignación de direcciones.

// VARIABLES ENTERAS

int z = 0;                    // Variable usada para incrementar las posiciones en los arreglos de direcciones.

```

Código 2.8. Declaración de variables enteras con y sin signo.

Para la realización del código en el archivo `usr_wireless.c` se modificaron las siguientes funciones según lo mostrado en los diagramas de flujo presentados en la sección 2.1 “DESCRIPCIÓN DEL ALGORITMO”.

- **Función `usr_wireless_app_task ()`**

Esta función debe ser editada por el usuario para agregar las primeras tareas de aplicación en el transceptor. Al energizar los transceptores todos empiezan como nodos por lo que la primera tarea es definir la dirección aleatoria y se envía una trama a sí mismo para visualizar dicha dirección.

En esta función también se añadieron las funciones relacionadas con el proceso de transmisión tanto para el proceso de asignación de direcciones como para el proceso de comunicación.

- **Función `usr_frame_received_cb ()`**

Esta función debe ser editada por el usuario para agregar las tareas cuando se recibe una trama. Aquí se realizan las tareas de procesamiento de tramas, por ejemplo, extraer el payload o algún elemento de una trama recibida.

- **Función `usr_frame_transmitted_cb ()`**

Esta función debe ser editada por el usuario para agregar tareas cuando se transmite un mensaje. Aquí los nodos asignadores envían la información necesaria a los nodos para que puedan calcular su nueva dirección fija y la definan.

Cabe indicar que esta función se utilizará para los tres criterios de asignación de direcciones. Como se observa en Código 2.9 la variable “dirInicial” empieza con el valor de 0x0000 y aumenta en 1, luego transforma esta variable de hexadecimal a cadena de caracteres, posteriormente almacena los bits disponibles que se calculan en `usr_app_timer_cb2` como se observa en Código 2.19, este proceso se realiza mientras se haya enviado la información a todos los nodos de los cuales el nodo asignador recibió respuestas, una vez hecho esto finaliza el proceso de asignación de direcciones para dicho nodo asignador. Mientras se realiza este proceso el nodo asignador también hace uso de la información que proporciona a los nodos para que calculen y definan su nueva dirección, con el fin de almacenar las direcciones de sus nodos rama.

```

if (direccionamiento) // Etapa de direccionamiento
{
    if (transmitirDirs) // Verifica el estado de la variable transmitirDirs.
    {
        dirInicial = dirInicial + 0x0001; // Se aumenta la dirInicial en un valor de 1 hexadecimal.
        itoa(dirInicial,nuevaDir,16); // Convierte el valor de la variable dirInicial de hexadecimal a texto.
        nuevaDir[1] = bitsDisponibles; // Se asigna en la posición 1 de la variable nuevaDir el valor de la variable bitsDisponibles.
        transmit_sample_frame((uint8_t*)nuevaDir,strlen(nuevaDir),dirNodos[1]); // Se transmite una trama unicast con la información
        // contenida en la variable nuevaDir hacia los nodos
        // que respondieron a la oferta de asignación de direcciones.

        delay_ms(50); // Retardo para transmitir.
        dirCalculadaCom = dirInicial << bitsDisponibles; // Desplaza los bits de la variable contador la cantidad de espacios que especifica
        // la variable numBits y se asigna ese valor a dirCalculada.
        dirComunicacion[z] = dirCalculadaCom + tal_pib.ShortAddress; // Se suma el valor de dirCalculadaCom más la dirección actual del
        // nodo y se asigna ese valor a dirComunicacion.

        z++; // Aumento el contador z.
        if (l == contNodos-1) // Verifica que el valor del contador l sea igual al valor de contNodos menos 1.
        {
            transmitirDirs = false; // Cambia el estado de la variable transmitirDirs.
            direccionamiento = false; // Cambia el estado de la variable direccionamiento.
            comunicacion = true; // Cambia el estado de la variable comunicacion.
            APAGAR_LED(LED1R); // Se apaga el LED Rojo.
            APAGAR_LED(LED3Y); // Se apaga el LED Amarillo.
            delay_ms(50); // Retardo para la siguiente verificación.
        }
        l++; // Aumento el contador l.
    }
}
}

```

Código 2.9. Función para enviar información a los nodos para que definan su dirección fija.

- **Función `usr_app_timer_cb1 ()`**

Esta función determina las siguientes tareas:

- ✓ Si el nodo solamente recibió la oferta de asignación de un nodo asignador le responde directamente.
- ✓ Si el nodo recibió la oferta de asignación de dos o más nodos asignadores responde según el criterio con el que se esté trabajando.

- **Función `usr_app_timer_cb2 ()`**

Esta función determina el número de bits usados y bits disponibles, así como llamar a la función `usr_frame_transmitted_cb ()` mediante la cual se enviará la información a los nodos para que calculen y definan su nueva dirección.

Para iniciar el proceso de asignación de direcciones, independientemente del criterio que se utilice, en primer lugar, se debe energizar los transceptores, al momento de realizar esto los nodos se envían a sí mismos una trama con el mensaje "MY_ID" para visualizar la dirección aleatoria que posee. En esta función como en todas las que se usan para transmisión se usa la instrucción `transmit_sample_frame()`, la cual permite transmitir una trama y necesita parámetros de entrada como: el mensaje del payload, la longitud del payload y la dirección destino.

```
if(dirAleatoriaNodo) // Verifica el estado de la variable dirAleatoriaNodo.
{
    dirAsignada = CCPU_ENDIAN_TO_LE16(tal_pib.ShortAddress); // Se asigna la dirección aleatoria del nodo a la variable dirAsignada.
    transmit_sample_frame((uint8_t*)"MY_ID",5,dirAsignada); // Se transmite una trama unicast con el mensaje "MY_ID"
    delay_ms(50); // Retardo para transmitir.
    dirAleatoriaNodo = false; // Cambia de estado la variable dirAleatoriaNodo.
}
```

Código 2.10. Función para visualizar la dirección aleatoria de los nodos.

Antes de dar inicio al proceso de asignación de direcciones se puede asignar a los nodos un estado según el escenario que se desea simular mediante la pulsación del botón SW2, la cual se reconoce mediante la instrucción `!iort_get_pin_level(GPIO_PUSH_BUTTON_0)`, por ejemplo, en caso de que se desee simular un enlace ruidoso la variable "enlaceRuidoso" deber ser igual a `true`, en caso de que se desee simular un escenario de nodo caído la variable "nodoCaído" debe ser igual a `true`, por último en el criterio de asignación de direcciones por número de saltos para probar el algoritmo la variable "escenarioSaltos" debe estar en `true`.

Para que se dé inicio al proceso de comunicación tanto la variable “comunicacion” como “transmitComunicacion” deben ser igual a *true*, además se debe definir una dirección de destino.

```

if (!ioport_get_pin_level(GPIO_PUSH_BUTTON_0)) // Verifica el estado del botón SW2.
{
    delay_ms(200); // Retardo para reconocer la presión del botón.
    cont++; // Aumento el contador de presión del botón.
    if (cont == 1) // Verifica si el contador es igual a 1, es decir, se presionó una vez el botón.
    {
        //TAREAS DE LOS NODOS
        enlaceRuidoso = false; // Simula el escenario enlace ruidoso según sea el caso de prueba.
        nodoCaído = false; // Simula el escenario nodo caído según sea el caso de prueba.
        comunicacion = false; // Activa o desactiva la etapa de comunicación.
        transmitComunicacion = false; //Activa o desactiva la transmisión en la etapa de comunicación.
        escenarioSaltos = false; // Simula el escenario de saltos en el criterio de selección de ruta por número de saltos.
        dirDestino = 0x0000; // Indica la dirección destino de la trama que envía el nodo que inicia la etapa de comunicación.
        ENCENDER_LED(LED3Y); // Se enciende el LED Amarillo para comprobar que se entró a esta función.
        delay_ms(50); // Retardo para visualizar el encendido del LED.
        APAGAR_LED(LED3Y); // Se apaga el LED Amarillo.
    }
}

```

Código 2.11. Función para asignar un estado a los nodos.

2.2.1 Codificación del algoritmo usando el criterio de selección de ruta por orden de llegada de oferta de asignación

Para dar inicio al proceso de asignación de direcciones se debe presionar dos veces el botón SW2, al hacer esto el nodo en el cual se pulsó el SW2 se convierte en Gateway y se define el criterio de selección de ruta por orden de llegada al poner la variable “ordenLlegada” en *true*, como se observa en el Código 2.12.

```

if (cont == 2)
{
    //TAREAS DEL GATEWAY

    gateway = true; // Verifica si el contador es igual a 1, es decir, se presionó dos veces el botón.
    rxRespNodo = true; // Se activa la opción para que los asignadores puedan recibir respuestas de los nodos.
    ordenLlegada = true; // Indica que el criterio de selección de ruta es por orden de llegada de la oferta de asignación.
    ordenPotencia = false; // Indica que el criterio de selección de ruta es por nivel de potencia de la trama recibida.
    ordenSaltos = false; // Indica que el criterio de selección de ruta es por número de saltos.
    tal_pib_set(macShortAddress,(pib_value_t *)&dirInicial); // Instrucción que permite modificar la dirección actual y definir una nueva.
}

```

Código 2.12. Función para inicio del proceso de direccionamiento por orden de llegada.

Cabe mencionar que como se explicó en la sección 2.1 “DESCRIPCIÓN DEL ALGORITMO” la variable “enlaceArreglado” hace referencia al caso en que se haya simulado el escenario de enlace ruidosos o nodos caídos, caso contrario se tratará de un escenario sin fallos.

A continuación, se explicará el proceso de asignación de direcciones en un escenario sin fallos.

Para un escenario sin fallos el Gateway envía una trama con dirección destino broadcast con un mensaje “L”.


```

transmit_sample_frame((uint8_t*)"L",1,dirBroadcast); // Se transmite la trama con el mensaje "L" en broadcast.
delay_ms(50); // Retardo para transmitir.

```

Código 2.13. Función para envío de mensaje en broadcast.

En la función `usr_frame_received_cb ()`, se blanquea la trama, se copia la MPDU (MAC Protocol Data Unit) a la trama que se declaró previamente y se libera el buffer en caso de que se haya recibido otras tramas, como se observa en el Código 2.14.

```

memset(&trama_rx,0,sizeof(trama_rx)); //Blanqueo la trama de recepción
memcpy(&trama_rx,frame->mpdu,sizeof(trama_rx)); //Copio la mpdu de la trama recibida en la trama creada trama_rx
bmm_buffer_free(frame->buffer_header); //Vacío el buffer para la siguiente recepción

```

Código 2.14. Procedimiento inicial al recibir una trama.

Cuando los nodos que se encuentran dentro de la zona de cobertura del Gateway reciben la trama de broadcast, verifican la dirección destino, luego aumentan el contador de los nodos asignadores de los cuales recibieron ofertas de asignación, se guarda la dirección del nodo asignador, se aumenta el contador y se inicia el timer1.

```

if (trama_rx.add_dest == dirBroadcast) // Verifica si la dirección destino de la trama recibida es broadcast (0xFFFF).
{
    contCoords ++; // Aumento el contador de asignadores.
    dirAsignadores[k] = trama_rx.add_origen; // Almaceno la dirección de los asignadores de los que recibio una trama
    k++; // Aumento el contador k.
    strncpy(payload_rx,trama_rx.carga,1); // Copio el primer valor del payload de la trama recibida y lo almaceno en
    // payload_rx.
    ENCENDER_LED(LED2G); // Se enciende el LED Verde.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    start_timer1(); // Inicio el timer 1.
}

```

Código 2.15. Procedimiento de los nodos al recibir una trama con dirección broadcast.

En caso de que el nodo haya recibido la oferta de asignación solamente de un nodo asignador le responde directamente a dicho nodo asignador, por lo que en la instrucción `transmit_sample_frame` se especifica la dirección destino como "trama_rx.add_origen" la cual hace referencia a la dirección origen de la cual se recibió la trama y el mensaje de respuesta "NEED_ID_L", además de indicar el número de caracteres del mensaje, en este caso 9.

```

if (strcmp(payload_rx,"L") == 0) // Verifica si el contenido de payload_rx es igual a "L".
{
    transmit_sample_frame((uint8_t*)"NEED_ID_L",9,trama_rx.add_origen); // Transmito una trama unicast con el mensaje "NEED_ID_L"
    // a la dirección origen de la trama recibida.
    delay_ms(50); // Retardo para transmitir.
    ENCENDER_LED(LED2G); // Se enciende el LED Verde.
    rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
    dirEnviada = true; // Cambia el estado de la variable dirEnviada.
}

```

Código 2.16. Procedimiento en caso de que sea un nodo asignador.

En caso de que el nodo haya recibido la oferta de asignación de más de un nodo asignador, responde al primer nodo asignador del cual almaceno la dirección en el arreglo “dirAsignadores[]”, por lo que en la instrucción transmit_sample_frame se especifica la dirección destino como “dirAsignadores[0]” y el mensaje de respuesta “NEED_ID_L”, además de indicar el número de caracteres del mensaje, en este caso 9.

```
if (strcmp(payload_rx,"L") == 0) // Verifica si el contenido de payload_rx es igual a "L".
{
    transmit_sample_frame((uint8_t*)"NEED_ID_L",9,dirAsignadores[0]); // Transmite una trama unicast con el mensaje "NEED_ID_L"
                                                                    // a la dirección origen de la trama recibida.
    delay_ms(50); // Retardo para transmitir.
    rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
    dirEnviada = true; // Cambia el estado de la variable dirEnviada.
}
```

Código 2.17. Procedimiento en caso de que sea más de un nodo asignador.

Cuando el nodo asignador recibe la o las respuestas de los nodos, aumenta el contador de nodos, almacena las direcciones de dichos nodos, aumenta el contador y da inicio al timer2.

```
contNodos ++; // Aumento el contador de nodos de los que se recibió tramas.
dirNodos[k] = trama_rx.add_origen; // Almaceno la dirección de los nodos que enviaron tramas.
k++; // Aumento el contador k.
ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
delay_ms(50); // Retardo para visualizar el encendido del LED.
start_timer2(); // Inicio el timer 2.
```

Código 2.18. Procedimiento del nodo asignador al recibir respuestas de los nodos.

La función usr_app_timer_cb2 permite a los nodos asignadores una vez contabilizados los nodos realizar los calculos de los bits que va a usar y los bits que seguirán disponibles para la asignación de direcciones. Luego de esto se llama a la función usr_frame_transmitted_cb () para realizar el procedimiento que se observa en Código 2.9.

```
rxRespNodo = false; // Cambia el estado de la variable rxRespNodo.
if (contNodos > 0 && contNodos <= 3) // Verifica si el valor de contNodos es mayor a 0 y menor o igual a 3.
{
    APAGAR_LED(LED1R); // Se apaga el LED Rojo.
    delay_ms(50); // Retardo para encender el LED.
    ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    bitsUsados = 0x02; // Se define el valor de 0x02 en la variable bitsUsados.
    bitsDisponibles = bitsDisponibles - bitsUsados; // Se calcula el valor de la variable bitsDisponibles.
    delay_ms(50); // Retardo para transmitir.
    transmitirDirs = true; // Cambia el estado de la variable transmitirDirs.
    usr_frame_transmitted_cb(MAC_SUCCESS,frame); // Se llama a la función que enviará la información de su nueva dirección
                                                // a los nodos.
}
if (contNodos > 3 && contNodos <= 7) // Verifica si el valor de contNodos es mayor a 3 y menor o igual a 7.
{
    APAGAR_LED(LED1R); // Se apaga el LED Rojo.
    delay_ms(50); // Retardo para encender el LED.
    ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    bitsUsados = 0x03; // Se define el valor de 0x03 en la variable bitsUsados.
    bitsDisponibles = bitsDisponibles - bitsUsados; // Se calcula el valor de la variable bitsDisponibles.
    delay_ms(50); // Retardo para transmitir.
    transmitirDirs = true; // Cambia el estado de la variable transmitirDirs.
    usr_frame_transmitted_cb(MAC_SUCCESS,frame); // Se llama a la función que enviará la información de su nueva dirección
                                                // a los nodos.
}
```

```

if (contNodos > 7 && contNodos <= 15) // Verifica si el valor de contNodos es mayor a 7 y menor o igual a 15.
{
    APAGAR_LED(LED1R); // Se apaga el LED Rojo.
    delay_ms(50); // Retardo para encender el LED.
    ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    bitsUsados = 0x04; // Se define el valor de 0x04 en la variable bitsUsados.
    bitsDisponibles = bitsDisponibles - bitsUsados; // Se calcula el valor de la variable bitsDisponibles.
    delay_ms(50); // Retardo para transmitir
    transmitirDir = true; // Cambia el estado de la variable transmitirDir.
    usr_frame_transmitted_cb(MAC_SUCCESS,frame); // Se llama a la función que enviará la información de su nueva dirección
                                                // a los nodos.
}

```

Código 2.19. Procedimiento del nodo asignador al recibir respuestas de los nodos.

Una vez realizado el procedimiento mencionado en Código 2.9, es decir, en la función `usr_frame_transmitted_cb ()` la variable “dirEnviada” se pone en *true*, lo que permite a los nodos recibir la tramas con la información necesaria para calcular y definir su nueva dirección fija, como se observa en Código 2.20.

Una vez que el nodo correspondiente recibe la información por parte del nodo asignador, almacena el número de bits disponibles que recibió en la variable “numBits”, luego entra al lazo `while` mientras la variable “asignacion” tenga un valor *true*, después compara la variable “temp” que posee un valor de 0x31 en hexadecimal que equivale a 1 en texto, si el primer valor del payload de la trama recibida es igual al valor de “temp” empieza el cálculo de la nueva dirección. Para dicho cálculo se toma la variable “contador” y se desplaza hacia la izquierda el número de bits disponibles correspondiente a “numBits” para almacenar este valor en “dirCalculada”, para obtener la dirección definitiva a “dirCalculada” se le suma el valor de la dirección de su nodo asignador, es decir, el valor de la variable “trama_rx.add_origen” y se almacena en la variable “dirAsignada”, para luego usar la instrucción `tal_pib_set(macShortAddress,(pib_value_t *)&dirAsignada)` la cual ayuda a modificar la dirección actual por el valor de “dirAsignada”.

Una vez definida la nueva dirección se cambia el valor de la variable “asignacion” a *false* para que se termine el lazo `while`, se reinician los valores de los contadores utilizados y este nodo pasa a ser nodo asignador al cambiar el valor de la variable “asignador” a *true*.

En caso de que el primer valor de la trama recibida no sea igual al valor de “temp”, se aumentan tanto el valor de la variable “contador” como “temp” en uno para posteriormente volver a realizar la comparación mencionada.

```

if (trama_rx.add_dest == dirAsignada && dirEnviada == true) // Verifica si la dirección destino de la trama recibida es igual a la
// dirAsignada, además verifica el estado de la variable dirEnviada.
{
    APAGAR_LED(LED2G); // Se apaga el LED Amarillo.
    numBits = trama_rx.carga[1]; // Se asigna el valor en la posición 1 del payload de la trama recibida a la variable numBits.
    asignacion = true; // Cambia el estado de la variable asignacion.
    while(asignacion){ // Inicia el lazo while mientras la variable asignacion tenga el estado true.
        if (trama_rx.carga[0] == temp) // Verifica si el valor de la posición 0 del payload de la trama recibida es igual al valor
// de la variable temp.
        {
            dirCalculada = contador << numBits; // Desplaza los bits de la variable contador la cantidad de espacios que especifica
// la variable numBits y se asigna ese valor a dirCalculada.
            dirAsignada = dirCalculada + trama_rx.add_origen; // Se suma el valor de dirCalculada más la dirección de origen de la
// trama recibida y se asigna ese valor a dirAsignada.
            tal_pib_set(macShortAddress,(pib_value_t *)&dirAsignada); // Instrucción que permite modificar la dirección actual y
// definir la dirAsignada.

            ENCENDER_LED(LED2G); // Se enciende el LED Verde.
            delay_ms(50); // Retardo para visualizar el encendido del LED.
            bitsDisponibles = numBits; // Se asigna el valor de la variable numBits a bitsDisponibles.
            asignacion = false; // Cambia el estado de la variable asignacion.
            dirEnviada = false; // Cambia el estado de la variable dirEnviada.
            asignador = true; // Cambia el estado de la variable asignador.
            rxRespNodo = true; // Cambia el estado de la variable rxRespNodo.
            contNodos = 0; // Se reinicia el valor del contNodos.
            k=0; // Se reinicia el valor de k.
            l=0; // Se reinicia el valor de l.
        }
        contador = contador + 0x0001; // Se aumenta el valor del contador.
        temp ++; // Se aumenta el valor de temp.
    }
}
}

```

Código 2.20. Procedimiento de los nodos para calcular y definir su nueva dirección fija.

Ya que el nodo que definió su dirección fija se convierte en nodo asignador realiza el mismo procedimiento del Gateway al iniciar el proceso de asignación de direcciones como se muestra en Código 2.13 y así sucesivamente se vuelve a realizar el mismo proceso hasta lo mencionado en Código 2.20.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de enlaces ruidosos.

Cuando inicia el proceso de asignación de direcciones se realiza lo mencionado en Código 2.13, si un nodo tiene la variable “enlaceRuidoso” en *true* según lo explicado en Código 2.11, entonces una vez que el Gateway envía la trama en broadcast dicho nodo no la procesa, luego espera un segundo y envía una trama unicast con el mensaje “FALLO” al nodo asignador que envió la trama que en primer lugar no procesó.

```

if (enlaceRuidoso) // Verifica el estado de la variable enlaceRuidoso.
{
    transmitEnlaceRuidoso = true; // Cambia el estado de la variable transmitEnlaceRuidoso.
    dirEnlaceRuidoso = trama_rx.add_origen; // Se asigna la dirección de origen de la trama recibida a la variable dirEnlaceRuidoso.
    enlaceRuidoso = false; // Cambia el estado de la variable enlaceRuidoso.
}

if (transmitEnlaceRuidoso) // Verifica el estado de la variable transmitEnlaceRuidoso.
{
    delay_ms(1000); // Retardo después de no procesar la primera trama enviado por parte del asignador.
    transmit_sample_frame((uint8_t*)"FALLO",5,dirEnlaceRuidoso); // Transmite una trama unicast con el mensaje "FALLO" al asignador del
//cual no procesó la trama la primera vez.
    delay_ms(50); // Retardo para transmitir.
    transmitEnlaceRuidoso = false; // Cambia de estado la variable transmitEnlaceRuidoso.
}

```

Código 2.21. Procedimiento del nodo al simular un escenario de enlace ruidoso.

Cuando el Gateway o un nodo asignador recibe la trama unicast con el mensaje “FALLO”, le responde al nodo con una trama unicast y el mensaje “L”, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```

if (strcmp(payload_tx,"L") == 0) // Verifica si el contenido de payload_tx es igual a "L".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenLlegada = true; // Cambia el estado de la variable ordenLlegada.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
                                                // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)"L",1,dirEnlaceArreglado); // Se transmite la trama con el mensaje "L" en unicast al nodo
                                                                // que presentó un escenario diferente al ambiente ideal.
    delay_ms(50); // Retardo para transmitir.
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}

```

Código 2.22. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de nodos caídos.

Luego de que el proceso de asignación de direcciones finaliza se puede simular el escenario de nodo caído, el cual es similar a la inserción de un nodo a la topología, si un nodo tiene la variable “nodoCaído” en *true* según lo explicado en Código 2.11, entonces el nodo envía una trama broadcast con el mensaje “CAIDO” a los nodos asignadores que se encuentran dentro de la zona de cobertura de dicho nodo.

```

if (nodoCaído) // Verifica el estado de la variable nodoCaído.
{
    transmit_sample_frame((uint8_t*)"CAIDO",5,dirBroadcast); // Transmite una trama broadcast con el mensaje "CAIDO".
    delay_ms(50); // Retardo para transmitir.
    nodoCaído = false; // Cambia de estado la variable nodoCaído.
}

```

Código 2.23. Procedimiento del nodo al simular un escenario de nodo caído.

Cuando el Gateway o un nodo asignador recibe la trama broadcast con el mensaje “CAIDO”, le responde al nodo con una trama unicast y el mensaje “L”, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```

if (strcmp(payload_tx,"L") == 0) // Verifica si el contenido de payload_tx es igual a "L".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenLlegada = true; // Cambia el estado de la variable ordenLlegada.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
    // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)"L",1,dirEnlaceArreglado); // Se transmite la trama con el mensaje "L" en unicast al nodo
    // que presentó un escenario diferente al ambiente ideal.
    delay_ms(50); // Retardo para transmitir.
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}

```

Código 2.24. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.

2.2.2 Codificación del algoritmo usando el criterio de selección de ruta por nivel de potencia de la trama recibida

Ciertas partes del algoritmo de la sección 2.2.1 son similares a las que se usaron para realizar el código con el criterio de nivel de potencia, por lo que solamente se mencionarán las partes que difieren con la sección 2.2.1.

Para dar inicio al proceso de asignación de direcciones se debe presionar dos veces el botón SW2, al hacer esto el nodo en el cual se pulsó el SW2 se convierte en Gateway y se define el criterio de selección de ruta por orden de potencia al poner la variable "ordenPotencia" en *true*, como se observa en el Código 2.25.

```

gateway = true; // Verifica si el contador es igual a 1, es decir, se presionó dos veces el botón.
rxRespNodo = true; // Se activa la opción para que los asignadores puedan recibir respuestas de los nodos.
ordenLlegada = false; // Indica que el criterio de selección de ruta es por orden de llegada de la oferta de asignación.
ordenPotencia = true; // Indica que el criterio de selección de ruta es por nivel de potencia de la trama recibida.
ordenSaltos = false; // Indica que el criterio de selección de ruta es por número de saltos.
tal_pib_set(macShortAddress,(pib_value_t *)&dirInicial); // Instrucción que permite modificar la dirección actual y definir una nueva.

```

Código 2.25. Función para inicio del proceso de direccionamiento por nivel de potencia.

A continuación, se explicará el proceso de asignación de direcciones en un escenario sin fallos.

Para un escenario sin fallos el Gateway envía una trama con dirección destino broadcast con un mensaje "P".

```

transmit_sample_frame((uint8_t*)"P",1,dirBroadcast); // Se transmite la trama con el mensaje "P" en broadcast.
delay_ms(50); // Retardo para transmitir.

```

Código 2.26. Función para envío de mensaje en broadcast.

El siguiente paso a seguir es el mismo que se mencionó en Código 2.14.

Cuando los nodos que se encuentran dentro de la zona de cobertura del Gateway reciben la trama de broadcast, verifican la dirección destino, después se extrae la potencia de la trama recibida y se la almacena en la variable “ed_value”, luego aumenta el contador de los nodos asignadores de los cuales recibieron ofertas de asignación, se guarda la dirección del nodo asignador, también se almacena el valor de “ed_value” en la variable “potencia” de la estructura “ruta”, finalmente se aumenta el contador y se inicia el timer1.

```

if (trama_rx.add_dest == dirBroadcast) // Verifica si la dirección destino de la trama recibida es broadcast (0xFFFF).
{
    rssi_base_val = tal_get_rssi_base_val(); // Obtiene el valor base del indicador de la fuerza de la señal recibida (RSSI).
    phy_frame_len = frame -> mpdu[0]; // Extrae la longitud de la trama
    ed_value = frame -> mpdu[phy_frame_len + LQI_LEN + ED_VAL_LEN] + rssi_base_val; // Obtiene el valor de potencia en dBm.
    contCoords ++; // Aumento el contador de asignadores.
    ruta.direccionCoord [k] = trama_rx.add_origen; // Almaceno la dirección de los asignadores de los que recibio una trama
                                                    // con el mensaje "P".

    ruta.potencia[k] = ed_value;
    k++; // Aumento el contador k.
    strncpy(payload_rx,trama_rx.carga,1); // Copio el primer valor del payload de la trama recibida y lo almaceno en
                                        // payload_rx.
    ENCENDER_LED(LED2G); // Se enciende el LED Verde.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    start_timer1(); // Inicio el timer 1.
}

```

Código 2.27. Procedimiento de los nodos al recibir una trama con dirección broadcast.

En caso de que el nodo haya recibido la oferta de asignación solamente de un nodo asignador le responde directamente a dicho nodo asignador, por lo que en la instrucción transmit_sample_frame se especifica la dirección destino como “trama_rx.add_origen” la cual hace referencia a la dirección origen de la cual se recibió la trama y el mensaje de respuesta “NEED_ID_P”, además de indicar el número de caracteres del mensaje, en este caso 9.

```

if (strcmp(payload_rx,"P") == 0) // Verifica si el contenido de payload_rx es igual a "P".
{
    transmit_sample_frame((uint8_t*)"NEED_ID_P",9,trama_rx.add_origen); // Transmito una trama unicast con el mensaje "NEED_ID_P"
                                                                    // a la dirección origen de la trama recibida.
    delay_ms(50); // Retardo para transmitir.
    ENCENDER_LED(LED2G); // Se enciende el LED Verde.
    rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
    dirEnviada = true; // Cambia el estado de la variable dirEnviada.
}

```

Código 2.28. Procedimiento en caso de que sea un nodo asignador.

En caso de que el nodo haya recibido la oferta de asignación de más de un nodo asignador, mediante un lazo for se recorre los valores tanto de dirección como de potencia de la estructura llamada “ruta”, mientras se recorre el lazo for se irá comparando los valores de potencia que se extrajeron de las tramas recibidas, al final se obtendrán las potencias ordenadas de mayor a menor con la respectiva dirección que les corresponde. Una vez finalizado el lazo for el primer elemento del arreglo de direcciones será el que posee la mayor potencia por lo que se responderá a dicha dirección con el mensaje

“NEED_ID_P”, con la longitud del mensaje (9) y la primera dirección del arreglo de la estructura.

```

if (strcmp(payload_rx,"P") == 0) // Verifica si el contenido de payload_rx es igual a "P".
{
    for (int i = 0; i < (contCoords-1); i++){ // Declaro el lazo for y se limita hasta el valor de contCoords menos 1.
        for(int j = i+1; j < contCoords ; j++){ // Declaro otro lazo for y se limita hasta el valor de contCoords.
            if(ruta.potencia[j] > ruta.potencia[i]){ // Condición para ordenar de mayor a menor.
                // Proceso de intercambio de valores hasta llegar al orden de mayor a menor.
                potComparada = ruta.potencia[j];
                dirComparada = ruta.direccionCoord[j];
                ruta.potencia[j] = ruta.potencia[i];
                ruta.direccionCoord[j] = ruta.direccionCoord[i];
                ruta.potencia[i] = potComparada;
                ruta.direccionCoord[i] = dirComparada;
            }
        }
    }
    transmit_sample_frame((uint8_t*)"NEED_ID_P",9,ruta.direccionCoord[0]); // Transmito una trama unicast con el mensaje "NEED_ID_P"
                                                                    // a la dirección del asignador del cual se recibió la
                                                                    // trama con mayor potencia.

    delay_ms(50); // Retardo para transmitir.
    rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
    dirEnviada = true; // Cambia el estado de la variable dirEnviada.
}

```

Código 2.29. Procedimiento en caso de que sea más de un nodo asignador.

Cuando el nodo asignador recibe la o las respuestas de los nodos, aumenta el contador de nodos, almacena las direcciones de dichos nodos, aumenta el contador y da inicio al timer2.

```

contNodos ++; // Aumento el contador de nodos de los que se recibió tramas.
dirNodos[k] = trama_rx.add_origen; // Almaceno la dirección de los nodos que enviaron tramas.
k++; // Aumento el contador k.
ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
delay_ms(50); // Retardo para visualizar el encendido del LED.
start_timer2(); // Inicio el timer 2.

```

Código 2.30. Procedimiento del nodo asignador al recibir respuestas de los nodos.

La función `usr_app_timer_cb2` como se observa en Código 2.19 permite a los nodos asignadores una vez contabilizados los nodos realizar los calculos de los bits que va a usar y los bits que seguirán disponibles para la asignación de direcciones. Luego de esto se llama a la función `usr_frame_transmitted_cb ()` para realizar el procedimiento que se observa en Código 2.9.

Una vez realizado el procedimiento mencionado en Código 2.9, es decir, en la función `usr_frame_transmitted_cb ()` la variable “dirEnviada” se pone en *true*, lo que permite a los nodos recibir la tramas con la información necesaria para calcular y definir su nueva dirección fija, como se observa en Código 2.20 con su respectiva explicación.

Ya que el nodo que definió su dirección fija se convierte en nodo asignador realiza el mismo procedimiento del Gateway al iniciar el proceso de asignación de direcciones como se muestra en Código 2.13 y así sucesivamente se vuelve a realizar el mismo proceso hasta lo mencionado en Código 2.20.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de enlaces ruidosos.

Cuando inicia el proceso de asignación de direcciones se realiza lo mencionado en Código 2.25, si un nodo tiene la variable “enlaceRuidoso” en *true* según lo explicado en Código 2.11, entonces una vez que el Gateway envía la trama en broadcast dicho nodo no la procesa, luego espera un segundo y envía una trama unicast con el mensaje “FALLO” al nodo asignador que envió la trama que en primer lugar no procesó.

```

if (enlaceRuidoso) // Verifica el estado de la variable enlaceRuidoso.
{
    transmitEnlaceRuidoso = true; // Cambia el estado de la variable transmitEnlaceRuidoso.
    dirEnlaceRuidoso = trama_rx.add_origen; // Se asigna la dirección de origen de la trama recibida a la variable dirEnlaceRuidoso.
    enlaceRuidoso = false; // Cambia el estado de la variable enlaceRuidoso.
}

if (transmitEnlaceRuidoso) // Verifica el estado de la variable transmitEnlaceRuidoso.
{
    delay_ms(1000); // Retardo después de no procesar la primera trama enviado por parte del asignador.
    transmit_sample_frame((uint8_t*)"FALLO",5,dirEnlaceRuidoso); // Transmite una trama unicast con el mensaje "FALLO" al asignador del
                                                                //cual no procesó la trama la primera vez.
    delay_ms(50); // Retardo para transmitir.
    transmitEnlaceRuidoso = false; // Cambia de estado la variable transmitEnlaceRuidoso.
}

```

Código 2.31. Procedimiento del nodo al simular un escenario de enlace ruidoso.

Cuando el Gateway o un nodo asignador recibe la trama unicast con el mensaje “FALLO”, le responde al nodo con una trama unicast y el mensaje “P”, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```

if (strcmp(payload_tx,"P") == 0) // Verifica si el contenido de payload_tx es igual a "P".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenPotencia = true; // Cambia el estado de la variable ordenPotencia.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
                                                // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)"P",1,dirEnlaceArreglado); // Se transmite la trama con el mensaje "P" en unicast al nodo
                                                                // que presentó un escenario diferente al ambiente ideal.
    delay_ms(50); // Retardo para transmitir
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}

```

Código 2.32. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de nodos caídos.

Luego de que el proceso de asignación de direcciones finaliza se puede simular el escenario de nodo caído, el cual es similar a la inserción de un nodo a la topología, si un

nodo tiene la variable “nodoCaído” en *true* según lo explicado en Código 2.11, entonces el nodo envía una trama broadcast con el mensaje “CAIDO” a los nodos asignadores que se encuentran dentro de la zona de cobertura de dicho nodo.

```
if (nodoCaído) // Verifica el estado de la variable nodoCaído.
{
    transmit_sample_frame((uint8_t*)"CAIDO",5,dirBroadcast); // Transmite una trama broadcast con el mensaje "CAIDO".
    delay_ms(50); // Retardo para transmitir.
    nodoCaído = false; // Cambia de estado la variable nodoCaído.
}
```

Código 2.33. Procedimiento del nodo al simular un escenario de nodo caído.

Cuando el Gateway o un nodo asignador recibe la trama broadcast con el mensaje “CAIDO”, le responde al nodo con una trama unicast y el mensaje “P”, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```
if (strcmp(payload_tx,"P") == 0) // Verifica si el contenido de payload_tx es igual a "P".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenPotencia = true; // Cambia el estado de la variable ordenPotencia.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
    // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)"P",1,dirEnlaceArreglado); // Se trasmite la trama con el mensaje "P" en unicast al nodo
    // que presentó un escenario diferente al ambiente ideal.
    delay_ms(50); // Retardo para transmitir
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}
```

Código 2.34. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.

2.2.3 Codificación del algoritmo usando el criterio de selección de ruta por número de saltos

Algunas de las partes del algoritmo de la sección 2.2.1 son similares a las que se usaron para realizar el código con el criterio por número de saltos, por lo que solamente se mencionarán las partes que difieren con la sección 2.2.1.

Para dar inicio al proceso de asignación de direcciones se debe presionar dos veces el botón SW2, al hacer esto el nodo en el cual se pulsó el SW2 se convierte en Gateway y se define el criterio de selección de ruta por número de saltos al poner la variable “ordenSaltos” en *true*, como se observa en el Código 2.35.

```

gateway = true; // Verifica si el contador es igual a 1, es decir, se presionó dos veces el botón.
rxRespModo = true; // Se activa la opción para que los asignadores puedan recibir respuestas de los nodos.
ordenLlegada = false; // Indica que el criterio de selección de ruta es por orden de llegada de la oferta de asignación.
ordenPotencia = false; // Indica que el criterio de selección de ruta es por nivel de potencia de la trama recibida.
ordenSaltos = true; // Indica que el criterio de selección de ruta es por número de saltos.
tal_pib_set(macShortAddress,(pib_value_t *)&dirInicial); // Instrucción que permite modificar la dirección actual y definir una nueva.

```

Código 2.35. Función para inicio del proceso de direccionamiento por número de saltos.

A continuación, se explicará el proceso de asignación de direcciones en un escenario sin fallos.

Para un escenario sin fallos el Gateway envía una trama con dirección destino broadcast con un mensaje “S”, junto con el número de saltos para llegar la Gateway, el cual se almacena en la variable “contSaltos”.

```

strcpy(payload_tx,"S"); // Copia el texto "S" en la variable payload_tx
strcpy(envioSaltos,"S"); // Copia el texto "S" en la variable envioSaltos
envioSaltos[1] = contSaltos; // Almacena el valor del contador de saltos en la posición 1 de la variable envioSaltos.

transmit_sample_frame((uint8_t*)envioSaltos,2,dirBroadcast); // Se transmite la trama con el contenido de envioSaltos
//en broadcast.
delay_ms(50); // Retardo para transmitir.

```

Código 2.36. Función para envío de mensaje en broadcast.

El siguiente paso a seguir es el mismo que se mencionó en Código 2.14.

Cuando los nodos que se encuentran dentro de la zona de cobertura del Gateway reciben la trama de broadcast, verifican la dirección destino, aumenta el contador de los nodos asignadores de los cuales recibieron ofertas de asignación, después se extrae el número de saltos para llegar al Gateway y se la almacena en la variable “contSaltos”, luego se guarda la dirección del nodo asignador, también se almacena el valor de “contSaltos” en la variable “saltos” de la estructura “rutaS”, finalmente se aumenta el contador y se inicia el timer1.

```

if (trama_rx.add_dest == dirBroadcast) // Verifica si la dirección destino de la trama recibida es broadcast (0xFFFF).
{
    contCoords++; // Aumento el contador de asignadores.
    contSaltos = trama_rx.carga[1]; // Asigno el valor situado en la posición 1 del payload de la trama recibida a la variable
    // contSaltos.
    rutaS.direccionCoord [k] = trama_rx.add_origen; // Almaceno la dirección de los asignadores de los que recibio una trama
    // con el mensaje "S".
    rutaS.saltos[k] = contSaltos; // Asigno el valor de la variable contSaltos a la variable saltos de la estructura rutaS.
    k++; // Aumento el contador k.
    strncpy(payload_rx,trama_rx.carga,1); // Copio el primer valor del payload de la trama recibida y lo almaceno en
    // payload_rx.
    ENCENDER_LED(LED2G); // Se enciende el LED Verde.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    start_timer1(); // Inicio el timer 1.
}

```

Código 2.37. Procedimiento de los nodos al recibir una trama con dirección broadcast.

En caso de que el nodo haya recibido la oferta de asignación solamente de un nodo asignador le responde directamente a dicho nodo asignador, por lo que en la instrucción

transmit_sample_frame se especifica la dirección destino como “trama_rx.add_origen” la cual hace referencia a la dirección origen de la cual se recibió la trama y el mensaje de respuesta “NEED_ID_S”, además de indicar el número de caracteres del mensaje, en este caso 9.

```
transmit_sample_frame((uint8_t*)"NEED_ID_S",9,trama_rx.add_origen); // Transmite una trama unicast con el mensaje "NEED_ID_S"
                                                                    // a la dirección origen de la trama recibida.
delay_ms(50); // Retardo para transmitir.
ENCENDER_LED(LED2G); // Se enciende el LED Verde.
rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
dirEnviada = true; // Cambia el estado de la variable dirEnviada.
```

Código 2.38. Procedimiento en caso de que sea un nodo asignador.

En caso de que el nodo haya recibido la oferta de asignación de más de un nodo asignador, mediante un lazo for se recorre los valores tanto de dirección como de número de saltos de la estructura llamada “rutaS”, mientras se recorre el lazo for se irá comparando los valores de número de saltos que se extrajeron de las tramas recibidas, al final se obtendrá el número de saltos ordenados de menor a mayor con la respectiva dirección que les corresponde. Una vez finalizado el lazo for el primer elemento del arreglo de direcciones será el que posee el menor número de saltos para llegar al Gateway por lo que se responderá a dicha dirección con el mensaje “NEED_ID_S”, con la longitud del mensaje (9) y la primera dirección del arreglo de la estructura.

```
if (strcmp(payload_rx,"S") == 0) // Verifica si el contenido de payload_rx es igual a "S".
{
    for (int i = 0; i < (contCoords-1); i++){ // Declaro el lazo for y se limita hasta el valor de contCoords menos 1.
        for(int j = i+1; j < contCoords; j++){ // Declaro otro lazo for y se limita hasta el valor de contCoords.
            if(rutaS.saltos[j] < rutaS.saltos[i]){ // Condición para ordenar de menor a mayor.
                // Proceso de intercambio de valores hasta llegar al orden de menor a mayor.
                saltosComparacion = rutaS.saltos[j];
                dirComparada = rutaS.direccionCoord[j];
                rutaS.saltos[j] = rutaS.saltos[i];
                rutaS.direccionCoord[j] = rutaS.direccionCoord[i];
                rutaS.saltos[i] = saltosComparacion;
                rutaS.direccionCoord[i] = dirComparada;
            }
        }
    }
    transmit_sample_frame((uint8_t*)"NEED_ID_S",9,rutaS.direccionCoord[0]); // Transmite una trama unicast con el mensaje "NEED_ID_P"
                                                                    // a la dirección del asignador del cual se necesite dar
                                                                    // menos saltos para llegar al gateway.
    delay_ms(50); // Retardo para transmitir.
    rxTramasAsignador = false; // Cambia el estado de la variable rxTramasAsignador.
    dirEnviada = true; // Cambia el estado de la variable dirEnviada.
}
```

Código 2.39. Procedimiento en caso de que sea más de un nodo asignador.

Cuando el nodo asignador recibe la o las respuestas de los nodos, aumenta el contador de nodos, almacena las direcciones de dichos nodos, aumenta el contador y da inicio al timer2.

```

contNodos ++; // Aumento el contador de nodos de los que se recibió tramas.
dirNodos[k] = trama_rx.add_origen; // Almaceno la dirección de los nodos que enviaron tramas.
k++; // Aumento el contador k.
ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
delay_ms(50); // Retardo para visualizar el encendido del LED.
start_timer2(); // Inicio el timer 2.

```

Código 2.40. Procedimiento del nodo asignador al recibir respuestas de los nodos.

La función `usr_app_timer_cb2` como se observa en Código 2.19 permite a los nodos asignadores una vez contabilizados los nodos realizar los calculos de los bits que va a usar y los bits que seguirán disponibles para la asignación de direcciones. Luego de esto se llama a la función `usr_frame_transmitted_cb ()` para realizar el procedimiento que se observa en Código 2.9.

Una vez realizado el procedimiento mencionado en Código 2.9, es decir, en la función `usr_frame_transmitted_cb ()` la variable “dirEnviada” se pone en *true*, lo que permite a los nodos recibir la tramas con la información necesaria para calcular y definir su nueva dirección fija, como se observa en Código 2.20 con su respectiva explicación.

Ya que el nodo que definió su dirección fija se convierte en nodo asignador realiza el mismo procedimiento del Gateway al iniciar el proceso de asignación de direcciones como se muestra en Código 2.13 y así sucesivamente se vuelve a realizar el mismo proceso hasta lo mencionado en Código 2.20.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de enlaces ruidosos.

Cuando inicia el proceso de asignación de direcciones se realiza lo mencionado en Código 2.35, si un nodo tiene la variable “enlaceRuidoso” en *true* según lo explicado en Código 2.11, entonces una vez que el Gateway envía la trama en broadcast dicho nodo no la procesa, luego espera un segundo y envía una trama unicast con el mensaje “FALLO” al nodo asignador que envió la trama que en primer lugar no procesó.

```

if (enlaceRuidoso) // Verifica el estado de la variable enlaceRuidoso.
{
    transmitEnlaceRuidoso = true; // Cambia el estado de la variable transmitEnlaceRuidoso.
    dirEnlaceRuidoso = trama_rx.add_origen; // Se asigna la dirección de origen de la trama recibida a la variable dirEnlaceRuidoso.
    enlaceRuidoso = false; // Cambia el estado de la variable enlaceRuidoso.
}

if (transmitEnlaceRuidoso) // Verifica el estado de la variable transmitEnlaceRuidoso.
{
    delay_ms(1000); // Retardo después de no procesar la primera trama enviado por parte del asignador.
    transmit_sample_frame((uint8_t*)"FALLO",5,dirEnlaceRuidoso); // Transmite una trama unicast con el mensaje "FALLO" al asignador del
    //cual no procesó la trama la primera vez.
    delay_ms(50); // Retardo para transmitir.
    transmitEnlaceRuidoso = false; // Cambia de estado la variable transmitEnlaceRuidoso.
}

```

Código 2.41. Procedimiento del nodo al simular un escenario de enlace ruidoso.

Cuando el Gateway o un nodo asignador recibe la trama unicast con el mensaje “FALLO”, le responde al nodo con una trama unicast y el mensaje “S” acompañado del número de saltos necesarios para llegar al Gateway, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```

if (strcmp(payload_tx,"S") == 0) // Verifica si el contenido de payload_tx es igual a "S".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenSaltos = true; // Cambia el estado de la variable ordenSaltos.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
    // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)envioSaltos,2,dirEnlaceArreglado); // Se transmite la trama con el contenido de envioSaltos
    //en unicast al nodo que presentó un escenario diferente
    // al ambiente ideal.
    delay_ms(50); // Retardo para transmitir
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}

```

Código 2.42. Procedimiento del Gateway o un nodo asignador al simular un escenario de enlace ruidoso.

A continuación, se explicará el proceso de asignación de direcciones en un escenario de nodos caídos.

Luego de que el proceso de asignación de direcciones finaliza se puede simular el escenario de nodo caído, el cual es similar a la inserción de un nodo a la topología, si un nodo tiene la variable “nodoCaído” en *true* según lo explicado en Código 2.11, entonces el nodo envía una trama broadcast con el mensaje “CAIDO” a los nodos asignadores que se encuentran dentro de la zona de cobertura de dicho nodo.

```

if (nodoCaído) // Verifica el estado de la variable nodoCaído.
{
    transmit_sample_frame((uint8_t*)"CAIDO",5,dirBroadcast); // Transmite una trama broadcast con el mensaje "CAIDO".
    delay_ms(50); // Retardo para transmitir.
    nodoCaído = false; // Cambia de estado la variable nodoCaído.
}

```

Código 2.43. Procedimiento del nodo al simular un escenario de nodo caído.

Cuando el Gateway o nodo asignador recibe la trama broadcast con el mensaje “CAIDO”, le responde al nodo con una trama unicast y el mensaje “S” acompañado del número de saltos necesarios para llegar al Gateway, para luego realizar el proceso de asignación de direcciones tal como se explica desde Código 2.14 hasta Código 2.20.

```

if (strcmp(payload_tx,"S") == 0) // Verifica si el contenido de payload_tx es igual a "S".
{
    contEnvioBroadcast = 0; // Reinicia el valor de contEnvioBroadcast.
    ordenSaltos = true; // Cambia el estado de la variable ordenSaltos.
    enlaceArreglado = true; // Cambia el estado de la variable enlaceArreglado.
    dirEnlaceArreglado = trama_rx.add_origen; // Asigna el valor de la dirección de origen de la trama recibida
    // a la variable dirEnlaceArreglado.
}

if (enlaceArreglado) // Verifica si el enlace es arreglado
{
    transmit_sample_frame((uint8_t*)envioSaltos,2,dirEnlaceArreglado); // Se transmite la trama con el contenido de envioSaltos
    //en unicast al nodo que presentó un escenario diferente
    // al ambiente ideal.
    delay_ms(50); // Retardo para transmitir
    enlaceArreglado = false; // Cambia el estado de la variable enlaceArreglado.
}

```

Código 2.44. Procedimiento del Gateway o un nodo asignador al simular un escenario de nodo caído.

2.2.4 Codificación del algoritmo de comunicación de extremo a extremo

Para el proceso de comunicación de extremo a extremo cabe mencionar que esto implica, tanto la comunicación entre un nodo y el Gateway o entre dos nodos de la red. Dado que el Gateway básicamente es otro nodo de la red se utilizó el mismo código para los dos casos que se mencionaron.

Una vez que ha finalizado el proceso de asignación de direcciones se puede dar inicio al proceso de comunicación, para ello como se mencionó antes del direccionamiento se debe configurar el transceptor para este proceso cambiando las variables “comunicacion” y “transmitComunicacion” en *true*.

```

if (cont == 1) // Verifica si el contador es igual a 1, es decir, se presionó una vez el botón.
{
    //TAREAS DE LOS NODOS
    enlaceRuidoso = false; // Simula el escenario enlace ruidoso según sea el caso de prueba.
    nodoCaído = false; // Simula el escenario nodo caído según sea el caso de prueba.
    comunicacion = true; // Activa o desactiva la etapa de comunicación.
    transmitComunicacion = true; //Activa o desactiva la transmisión en la etapa de comunicación.
    escenarioSaltos = false; // Simula el escenario de saltos en el criterio de selección de ruta por número de saltos.
    dirDestino = 0x0000; // Indica la dirección destino de la trama que envía el nodo que inicia la etapa de comunicación.
    ENCENDER_LED(LED3Y); // Se enciende el LED Amarillo para comprobar que se entró a esta función.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    APAGAR_LED(LED3Y); // Se apaga el LED Amarillo.
}

```

Código 2.45. Configuración de los nodos para el proceso de comunicación.

Dependiendo del criterio que se haya utilizado para el proceso de asignación de direcciones, al iniciar el proceso de comunicación se enviará el mensaje destino que consta de la palabra “ENVIO” acompañada de la dirección destino, por ejemplo, la variable “mensajeDestino” tendrá el valor “ENVIO0000”, en caso de que la dirección destino no sea la del nodo actual, este envía dicho mensaje a la dirección de su nodo asignador.

```

if (transmitComunicacion) // Verifica el estado de la variable transmitComunicacion.
{
    itoa(dirDestino,dirDestinoConvertida,16); // Convierte el valor de la variable dirDestino de hexadecimal a texto.
    strcpy(mensajeDestino,"ENVIO"); // Copia el texto "ENVIO" en la variable mensajeDestino.
    strcat(mensajeDestino,dirDestinoConvertida); // Concatena el contenido de la variable dirDestinoConvertida al
                                                // contenido de la variable mensajeDestino.
    if ((strcmp(payload_tx,"L") == 0)) // Verifica si el contenido de payload_tx es igual a "L".
    {
        if (dirDestino != tal_pib.ShortAddress) // Verifica si el valor de dirDestino es diferente a la dirección del nodo.
        {
            transmit_sample_frame((uint8_t*)mensajeDestino,9,dirAsignadores[0]); // Transmite una trama unicast a la dirección de su
                                                                                // asignador con el contenido de la variable mensajeDestino.

            delay_ms(50); // Retardo para transmitir.
            ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
            delay_ms(50); // Retardo para visualizar el encendido del LED.
            APAGAR_LED(LED1R); // Se apaga el LED Rojo.
            transmitComunicacion = false; // Cambia el estado de la variable transmitComunicacion.
        }
    }

    if(strcmp(payload_tx,"P") == 0) // Verifica si el contenido de payload_tx es igual a "P".
    {
        if (dirDestino != tal_pib.ShortAddress) // Verifica si el valor de dirDestino es diferente a la dirección del nodo.
        {
            transmit_sample_frame((uint8_t*)mensajeDestino,9,ruta.direccionCoord[0]); // Transmite una trama unicast a la dirección de su
                                                                                // asignador con el contenido de la variable mensajeDestino.

            delay_ms(50); // Retardo para transmitir.
            ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
            delay_ms(50); // Retardo para visualizar el encendido del LED.
            APAGAR_LED(LED1R); // Se apaga el LED Rojo.
            transmitComunicacion = false; // Cambia el estado de la variable transmitComunicacion.
        }
    }

    if(strcmp(payload_tx,"S") == 0) // Verifica si el contenido de payload_tx es igual a "S".
    {
        if (dirDestino != tal_pib.ShortAddress) // Verifica si el valor de dirDestino es diferente a la dirección del nodo.
        {
            transmit_sample_frame((uint8_t*)mensajeDestino,9,rutaS.direccionCoord[0]); // Transmite una trama unicast a la dirección de su
                                                                                // asignador con el contenido de la variable mensajeDestino.

            delay_ms(50); // Retardo para transmitir.
            ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
            delay_ms(50); // Retardo para visualizar el encendido del LED.
            APAGAR_LED(LED1R); // Se apaga el LED Rojo.
            transmitComunicacion = false; // Cambia el estado de la variable transmitComunicacion.
        }
    }
}
}
}

```

Código 2.46. Proceso de transmisión en la comunicación.

Cuando se recibe un mensaje en el proceso de comunicación, se separan los primeros cinco caracteres del payload correspondiente a la palabra "ENVIO", una vez que se validó esto, se toman los siguientes caracteres del payload a partir del sexto para obtener la dirección de destino en texto, luego mediante strtol() se transforma a un valor en hexadecimal y se almacena en la variable "dirRecibida".

Posteriormente se ingresa a un lazo for para verificar si la dirección destino corresponde a alguno de los nodos rama del nodo actual, en caso de que alguna dirección si corresponda se transmite un mensaje a dicho nodo rama como se observa en Código 2.48, para el caso en el que no corresponda ninguna dirección se regresa al proceso de transmisión mencionado en Código 2.46.


```

if (strcmp(mensajeCom,"ENVIO") == 0) // Verifica si el contenido de mensajeRx es igual a "ENVIO".
{
    strncpy(dirPayload,&trama_rx.carga[5],4); // Copio los últimos 4 caracteres del payload de la trama recibida en la
// variable dirPayload.
    dirPayload [4] = '\0'; // Límite la variable dirPayload con el carácter 0 al final.
    dirRecibida = strtol(dirPayload,NULL,16); // Transformo el valor de dirPayload que está en texto a valor hexadecimal
// y lo asigno a la variable dirRecibida.
    dirDestino = dirRecibida; // Asigno el valor de dirRecibida a dirDestino.
    for (int i = 0; i < z; i++) // Declaro un lazo for limitado hasta el valor de z.
    {
        if (dirRecibida == dirComunicacion[i]) // Verifica si dirRecibida es igual a alguna dirección del arreglo de
// direcciones llamado dirComunicacion.
        {
            esRama = true; // Cambia el estado de la variable esRama.
            dirRama = dirComunicacion[i]; // Asigna el valor de la dirección que coincidió a la variable dirRama.
            z = -1; // Se le asigna el valor de -1 a z para que no se vuelva a ingresar al lazo for.
            ENCENDER_LED(LED2G); // Se enciende el LED Verde.
        }
    }

    if (esRama) // Verifica que el estado de la variable esRama sea true.
    {
        transmitRama = true; // Cambia el estado de la variable transmitRama.
        delay_ms(50); // Retardo para transmitir.
    }
    else // Caso contrario si el estado de la variable esRama es false.
    {
        ENCENDER_LED(LED3Y); // Se enciende el LED Amarillo.
        transmitComunicacion = true; // Cambia el estado de la variable transmitComunicacion.
        delay_ms(50); // Retardo para transmitir.
    }
}
}

```

Código 2.47. Proceso de recepción en la comunicación.

```

if (transmitRama) // Verifica el estado de la variable transmitRama
{
    itoa(dirDestino,dirDestinoConvertida,16); // Convierte el valor de la variable dirDestino de hexadecimal a texto.
    strcpy(mensajeDestino,"ENVIO"); // Copia el texto "ENVIO" en la variable mensajeDestino.
    strcat(mensajeDestino,dirDestinoConvertida); // Concatena el contenido de la variable dirDestinoConvertida al
// contenido de la variable mensajeDestino.
    transmit_sample_frame((uint8_t*)mensajeDestino,9,dirRama); // Transmite una trama unicast a la dirección del nodo rama
// con el contenido de la variable mensajeDestino.

    delay_ms(50); // Retardo para transmitir
    ENCENDER_LED(LED1R); // Se enciende el LED Rojo.
    delay_ms(50); // Retardo para visualizar el encendido del LED.
    APAGAR_LED(LED1R); // Se apaga el LED Rojo.
    transmitRama = false; // Cambia el estado de la variable transmitRama
}
}

```

Código 2.48. Transmisión a un nodo al cual le asignó la dirección.

3. RESULTADOS Y DISCUSIÓN

En este capítulo se van a describir los resultados de las pruebas realizadas de los algoritmos desarrollados al ser implementados en una red inalámbrica de sensores, dentro de las pruebas que se realizaron están: proceso de asignación de direcciones con criterio de orden de llegada, nivel de potencia y número de saltos, cada uno de estos criterios se probaron en escenarios sin fallos, con enlaces ruidosos y con nodos caídos.

Para la visualización de los resultados se utilizará el Sniffer CC2531 Dongle y el software de captura de tramas Texas Instruments SmartRF Packet Sniffer. El SmartRF Packet Sniffer es una aplicación de software para PC que puede mostrar y almacenar paquetes de radio capturados por un dispositivo de RF que escucha. El dispositivo de captura está conectado a la PC a través de USB. Se admiten varios protocolos de RF. El Sniffer filtra, decodifica paquetes y muestra de una manera conveniente [17].

El prototipo está formado por 6 transceptores ATZB-256RFR2-XPRO y 1 transceptor RCB256RFR2, además del Sniffer mencionado anteriormente. El código de los algoritmos va a ser programados en los transceptores y se procederá a realizar las pruebas necesarias.

Los resultados que se espera obtener de las pruebas es comprobar el funcionamiento del algoritmo, es decir, que al final del proceso de asignación de direcciones todos los nodos se encuentren correctamente identificados, además de medir el tiempo que los nodos se demoran en generar la tabla de enrutamiento y también se medirá el tiempo que le toma a la red recuperarse de un escenario de enlaces ruidosos o nodos caídos.

Para armar el prototipo de red se colocaron los transceptores tanto ATZB-256RFR2-XPRO como RCB256RFR2 arriba de unos conos como se muestra en la Figura 3.1 y Figura 3.2 respectivamente. Además, el Sniffer se colocó bastante cerca de la red con el fin de que no existan errores al capturar las tramas y a su vez se conectó a la laptop mediante el conector USB, como se observa en la Figura 3.3.



Figura 3.1. Ubicación del transceptor ATZB-256RFR2-XPRO en el cono.



Figura 3.2. Ubicación del transceptor RCB256RFR2 en el cono.

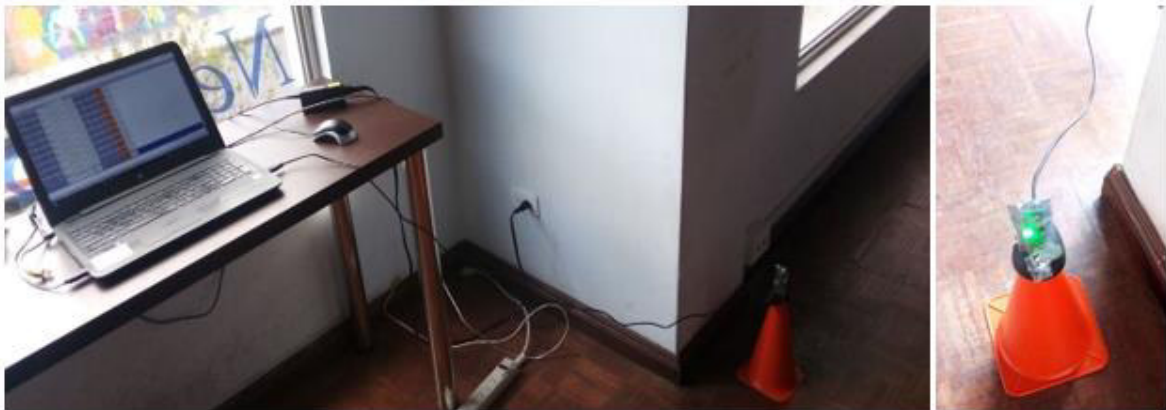


Figura 3.3. Sniffer conectado a la laptop.

3.1. PRUEBA 1

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.4, cabe mencionar que la distancia entre el Gateway y el primer nivel es de 2 m, la distancia entre los nodos del primer nivel es 1.45 m, la distancia entre los nodos del segundo y tercer nivel es 2 m y por último la distancia entre los nodos del tercer nivel es 1.45 m, en esta prueba se presenta un escenario sin fallos, el criterio para el proceso de asignación de direcciones será por orden de llegada de oferta de asignación, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.5.



Figura 3.4. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
1	=0	16	DATA	0	0	0	1	0x00	0xCAFE	0x6567	0x6567	102	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4989283		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
2	=4989283	16	DATA	0	0	0	1	0x00	0xCAFE	0x0D97	0x0D97	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6064447		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
3	=11053730	16	DATA	0	0	0	1	0x00	0xCAFE	0x4FC5	0x4FC5	120	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5855988		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
4	=16909718	16	DATA	0	0	0	1	0x00	0xCAFE	0x3602	0x3602	112	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6299455		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
5	=23209173	16	DATA	0	0	0	1	0x00	0xCAFE	0x4B1E	0x4B1E	142	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3774318		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
6	=26983491	16	DATA	0	0	0	1	0x00	0xCAFE	0x0D51	0x0D51	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5239746		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
7	=32223237	16	DATA	0	0	0	1	0x00	0xCAFE	0x27F6	0x27F6	93	OK

Figura 3.5. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje "L", luego los tres nodos correspondientes al segundo nivel responden al Gateway con el mensaje "NEED_ID_L", en el siguiente orden primero responde el nodo con la dirección 0x0D97, luego el nodo con la dirección 0x3602 y finalmente responde el nodo con la dirección 0x4FC5. De acuerdo con el orden de las tramas de la Figura 3.6, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que definan su dirección fija los nodos con la dirección 0x0D97 y 0x3602, dado que el nodo con la dirección 0x0D97 calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje "L", luego el nodo con la dirección 0x4FC5 recibe la información del Gateway, posteriormente el nodo

con dirección 0x3602 define su nueva dirección con el valor 0x4000 y envía una trama en broadcast con el mensaje “L”, después el nodo con dirección 0x4FC5 define su nueva dirección con el valor 0x6000 y envía una trama en broadcast con el mensaje “L”.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 8	+11856058 =44079295	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0000	L	110	OK
RX 9	+3063295 =47142590	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x0D97	NEED_ID_L	126	OK
RX 10	+47145768 =47145768	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x3602	NEED_ID_L	128	OK
RX 11	+15376 =47161144	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x4FC5	NEED_ID_L	174	OK
RX 12	+3330508 =50491652	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x0D97	0x0000	1	126	OK
RX 13	+85297 =50576949	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x3602	0x0000	2	126	OK
RX 14	+1109 =50578058	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x2000	L	126	OK
RX 15	+82628 =50660686	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x4FC5	0x0000	3	126	OK
RX 16	+2742 =50663428	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4000	L	134	OK
RX 17	+83507 =50746935	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x6000	L	174	OK

Figura 3.6. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.7 que los nodos del tercer nivel con las direcciones 0x0D51 y 0x27F6 respondieron con el mensaje “NEED_ID_L” al nodo asignador con la dirección 0x4000, luego el nodo con la dirección 0x4B1E le responde con el mensaje “NEED_ID_L” al nodo asignador con la dirección 0x6000, después el nodo asignador con la dirección 0x4000 envía información para que los nodos con dirección 0x0D51 y 0x27F6 calculen y definan su nueva dirección, hecho esto el nodo con dirección 0x0D51 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “L”, luego el nodo asignador con la dirección 0x6000 envía información para que el nodo con dirección 0x4B1E calcule y defina su nueva dirección, seguidamente el nodo con dirección 0x27F6 define su nueva dirección con el valor 0x5000 y envía una trama en broadcast con el mensaje “L”, finalmente el nodo con

dirección 0x4B1E define su nueva dirección con el valor 0x6800 y envía una trama en broadcast con el mensaje “L”.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 18	+3006738 =53753673	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4000	0x0D51	NEED_ID_L	99	OK
RX 19	+12657 =53766330	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4000	0x27F6	NEED_ID_L	59	OK
RX 20	+125130 =53891460	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x6000	0x4B1E	NEED_ID_L	139	OK
RX 21	+3013267 =56904727	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x0D51	0x4000	1 *	134	OK
RX 22	+82250 =56986977	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x27F6	0x4000	2 *	134	OK
RX 23	+3895 =56990872	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4800	L	99	OK
RX 24	+69505 =57060377	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x4B1E	0x6000	1 *	174	OK
RX 25	+12209 =57072586	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x5000	L	61	OK
RX 26	+84589 =57157175	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x6800	L	139	OK

Figura 3.7. Asignación de direcciones de los nodos asignadores.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.8, además una vez finalizado este proceso de acuerdo con la Figura 3.7 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 57.16 segundos.

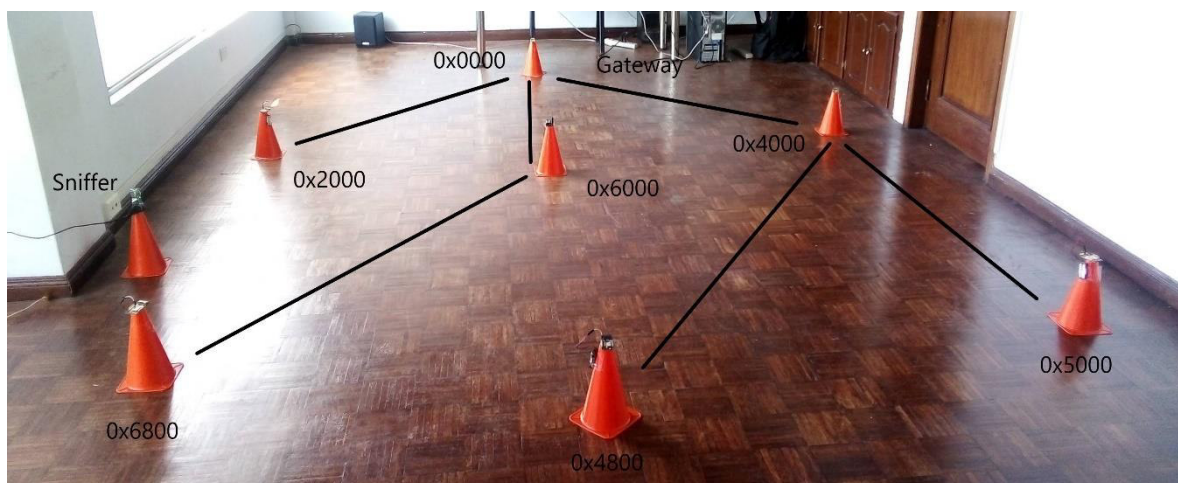


Figura 3.8. Topología final.

3.2. PRUEBA 2

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.9, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 1, en esta prueba se presenta un escenario con enlace ruidoso, el criterio para el proceso de asignación de direcciones será por orden de llegada de oferta de asignación, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.10. Dado que es un escenario con enlace ruidoso se debe mencionar que el nodo configurado para simular dicho escenario es el que tiene la dirección aleatoria 0x16D4, como se explicó en Código 2.11 en el nodo mencionado la variable “enlaceRuidoso” tiene el valor de *true*.

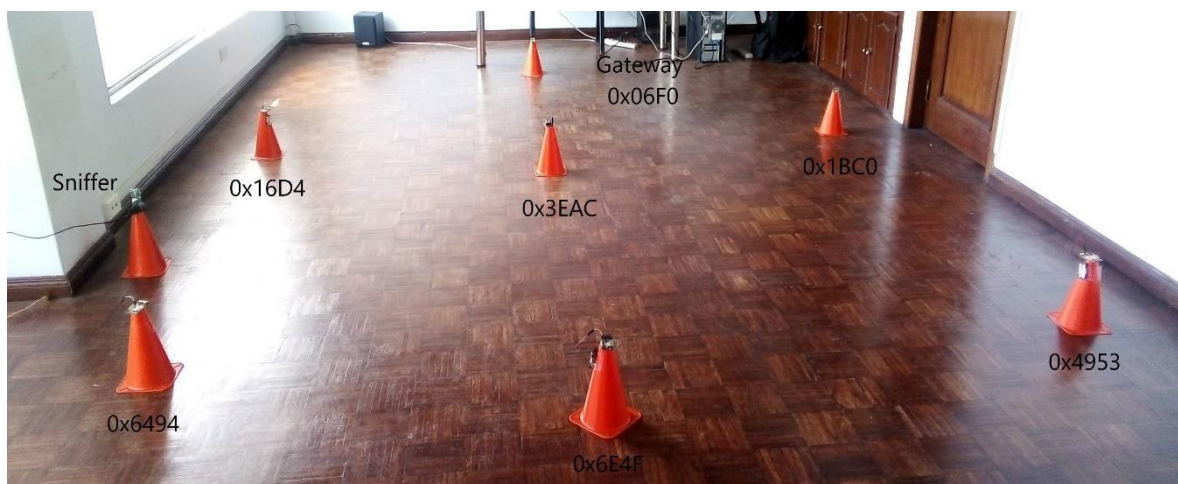


Figura 3.9. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
1	=0	16	DATA	0	0	0	1	0x00	0xC9FE	0x06F0	0x06F0	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4151928		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
2	=4151928	16	DATA	0	0	0	1	0x00	0xC9FE	0x16D4	0x16D4	131	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+8059770		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
3	=12211698	16	DATA	0	0	0	1	0x00	0xC9FE	0x3EAC	0x3EAC	112	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4133486		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
4	=16345184	16	DATA	0	0	0	1	0x00	0xC9FE	0x1BC0	0x1BC0	104	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4753162		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
5	=21098346	16	DATA	0	0	0	1	0x00	0xC9FE	0x6494	0x6494	139	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3521402		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
6	=24619748	16	DATA	0	0	0	1	0x00	0xC9FE	0x6E4F	0x6E4F	128	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3731564		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
7	=28351312	16	DATA	0	0	0	1	0x00	0xC9FE	0x4953	0x4953	85	OK

Figura 3.10. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones, el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “L”, luego como se mencionó en la sección 2.1.1.1 en el caso de enlaces ruidosos el nodo configurado en este escenario, es decir, el nodo con la dirección 0x16D4 no procesa la primera trama y espera 1 segundo para enviar una trama unicast con el mensaje “FALLO” al Gateway, como se observa en la trama 9 de la Figura 3.10 el nodo envía la trama después de alrededor de un segundo y medio, después el Gateway responde al nodo con una trama unicast y el mensaje “L”, posteriormente dado que en primer lugar el Gateway ya envió una trama broadcast con el mensaje “L” los tres nodos correspondientes al segundo nivel le responden con el mensaje “NEED_ID_L”, en el siguiente orden primero responde el nodo con la dirección 0x1BC0, luego el nodo con la dirección 0x3EAC y finalmente responde el nodo con la dirección 0x16D4. De acuerdo con el orden de las tramas de la Figura 3.10, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que definan su dirección fija los nodos con la dirección 0x1BC0 y 0x3EAC, dado que el nodo con la dirección 0x1BC0 calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “L”, seguidamente dado que el nodo con la dirección 0x3EAC calcula y define su nueva dirección con el valor de 0x4000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “L”, finalmente el nodo con la dirección 0x16D4 recibe la información del Gateway define su nueva dirección con el valor 0x6000 y envía una trama en broadcast con el mensaje “L”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+13725312		Type	Sec	Pnd	Ack.req	PAN_compr				L			
8	=42076624	12	DATA	0	0	0	1	0x01	0xC9FE	0xFFFF	0x0000	96	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1659622		Type	Sec	Pnd	Ack.req	PAN_compr				FAL			
9	=43736246	16	DATA	0	0	0	1	0x01	0xC9FE	0x0000	0x16D4	126	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+3586		Type	Sec	Pnd	Ack.req	PAN_compr				L			
10	=43739832	12	DATA	0	0	0	1	0x02	0xC9FE	0x16D4	0x0000	115	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1404248		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L			
11	=45144080	20	DATA	0	0	0	1	0x01	0xC9FE	0x0000	0x1BC0	123	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+9362		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L			
12	=45153442	20	DATA	0	0	0	1	0x01	0xC9FE	0x0000	0x3EAC	171	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1725455		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L			
13	=46878897	20	DATA	0	0	0	1	0x02	0xC9FE	0x0000	0x16D4	120	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1612670		Type	Sec	Pnd	Ack.req	PAN_compr				1			
14	=48491567	13	DATA	0	0	0	1	0x03	0xC9FE	0x1BC0	0x0000	*	112	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+82682		Type	Sec	Pnd	Ack.req	PAN_compr				2			
15	=48574249	13	DATA	0	0	0	1	0x04	0xC9FE	0x3EAC	0x0000	*	112	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+3770		Type	Sec	Pnd	Ack.req	PAN_compr				L			
16	=48578019	12	DATA	0	0	0	1	0x02	0xC9FE	0xFFFF	0x2000	123	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+81119		Type	Sec	Pnd	Ack.req	PAN_compr				L			
17	=48659138	12	DATA	0	0	0	1	0x02	0xC9FE	0xFFFF	0x4000	171	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1014		Type	Sec	Pnd	Ack.req	PAN_compr				3			
18	=48660152	13	DATA	0	0	0	1	0x05	0xC9FE	0x16D4	0x0000	*	112	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+86357		Type	Sec	Pnd	Ack.req	PAN_compr				L			
19	=48746509	12	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	0x6000	120	OK	

Figura 3.11. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.11 el nodo de tercer nivel con la dirección 0x6494 le responde con el mensaje "NEED_ID_L" al nodo asignador con la dirección 0x4000, luego el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x6494 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x6494 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje "L", seguidamente los nodos del tercer nivel con las direcciones 0x6E4F y 0x4953 respondieron con el mensaje "NEED_ID_L" al nodo asignador con la dirección 0x4800, después el nodo asignador con la dirección 0x4800 envía información para que los nodos con dirección 0x6E4F y 0x4953 calculen y definan su nueva dirección, luego el nodo con dirección 0x6E4F define su nueva dirección con el valor 0x4A00 y envía una trama en broadcast con el mensaje "L", finalmente el nodo con dirección 0x4953 define su nueva dirección con el valor 0x4C00 y envía una trama en broadcast con el mensaje "L".

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3057326		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L	136	OK
20	=51803835	20	DATA	0	0	0	1	0x01	0xCAFE	0x4000	0x6494		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3159990		Type	Sec	Pnd	Ack.req	PAN_compr				1	171	OK
21	=54963825	13	DATA	0	0	0	1	0x03	0xCAFE	0x6494	0x4000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+88580		Type	Sec	Pnd	Ack.req	PAN_compr				L	136	OK
22	=55052405	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4800		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3086559		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L	91	OK
23	=58138964	20	DATA	0	0	0	1	0x01	0xCAFE	0x4800	0x6E4F		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+13594		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L	88	OK
24	=58152558	20	DATA	0	0	0	1	0x01	0xCAFE	0x4800	0x4953		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3214104		Type	Sec	Pnd	Ack.req	PAN_compr				1	136	OK
25	=61366662	13	DATA	0	0	0	1	0x03	0xCAFE	0x6E4F	0x4800	-	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+85564		Type	Sec	Pnd	Ack.req	PAN_compr				2	136	OK
26	=61452226	13	DATA	0	0	0	1	0x04	0xCAFE	0x4953	0x4800	-	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4177		Type	Sec	Pnd	Ack.req	PAN_compr				L	75	OK
27	=61456403	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4A00		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+82643		Type	Sec	Pnd	Ack.req	PAN_compr				L	96	OK
28	=61539046	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4C00		

Figura 3.12. Asignación de direcciones de los nodos asignadores.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.13, además una vez finalizado este proceso de acuerdo con la Figura 3.12 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 61.54 segundos.

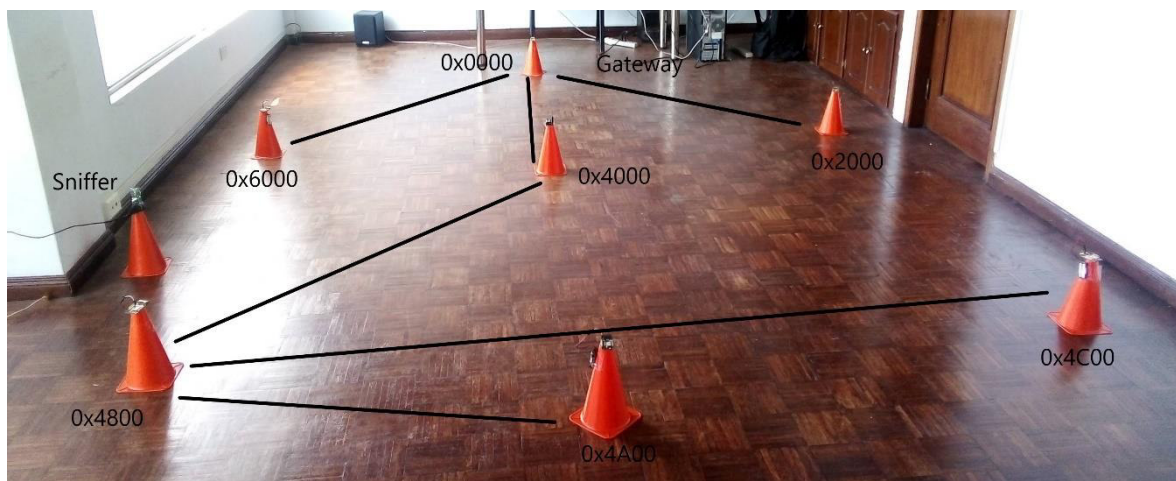


Figura 3.13. Topología final.

3.3. PRUEBA 3

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.14, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 1, en esta prueba se presenta un escenario con nodo caído, el criterio para el proceso de asignación de direcciones será por orden de llegada de oferta de asignación, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.15. Dado que es un escenario con nodo caído se debe mencionar que el nodo configurado para simular dicho escenario es el que no tiene dirección aleatoria en la Figura 3.14 ya que al inicio se encuentra apagado y una vez finalizado el proceso de asignación de direcciones se encendió manualmente para probar el escenario con nodo caído, como se explicó en Código 2.11 en el nodo mencionado la variable “nodoCaído” tiene el valor de *true*.



Figura 3.14. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+0	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
1	=0	16	DATA	0	0	0	1		0x00	0xCAFE	0x0186	0x0186	ID	118	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+5391863	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
2	=5391863	16	DATA	0	0	0	1		0x00	0xCAFE	0x1C5C	0x1C5C	ID	120	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+5114834	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
3	=10506697	16	DATA	0	0	0	1		0x00	0xCAFE	0x57FB	0x57FB	ID	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+4845085	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
4	=15351782	16	DATA	0	0	0	1		0x00	0xCAFE	0x2A67	0x2A67	ID	118	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+5557403	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
5	=20909185	16	DATA	0	0	0	1		0x00	0xCAFE	0x0D2E	0x0D2E	ID	142	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS		
RX	+2963005	16	Type	Sec	Pnd	Ack.req	PAN	compr			MY_				
6	=23872190	16	DATA	0	0	0	1		0x00	0xCAFE	0x4E10	0x4E10	ID	131	OK

Figura 3.15. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “L”, luego los tres nodos correspondientes al segundo nivel responden al Gateway con el mensaje “NEED_ID_L”, en el siguiente orden primero responde el nodo con la dirección 0x1C5C, luego el nodo con la dirección 0x57FB y finalmente responde el nodo con la dirección 0x2A67. De acuerdo con el orden de las tramas de la Figura 3.16, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que definan su dirección fija los nodos con la dirección 0x1C5C y 0x57FB, dado que el nodo con la dirección 0x1C5C calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “L”, seguidamente el nodo con la dirección 0x57FB calcula y define su nueva dirección con el valor de 0x4000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “L”, luego el nodo con la dirección 0x2A67 recibe la información del Gateway, posteriormente el nodo con dirección 0x2A67 define su nueva dirección con el valor 0x6000 y envía una trama en broadcast con el mensaje “L”.

P.nbr. RX 7	Time (us) +13064460 =36936650	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x0000	MAC payload L	LQI 118	FCS OK
P.nbr. RX 8	Time (us) +3060205 =39996855	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x1C5C	MAC payload NEED_ ID_L	LQI 123	FCS OK
P.nbr. RX 9	Time (us) +14588 =40011443	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x57FB	MAC payload NEED_ ID_L	LQI 171	FCS OK
P.nbr. RX 10	Time (us) +3248 =40014691	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x2A67	MAC payload NEED_ ID_L	LQI 118	FCS OK
P.nbr. RX 11	Time (us) +3330013 =43344704	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x1C5C	Source Address 0x0000	MAC payload 1 *	LQI 120	FCS OK
P.nbr. RX 12	Time (us) +82978 =43427682	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x57FB	Source Address 0x0000	MAC payload 2 *	LQI 120	FCS OK
P.nbr. RX 13	Time (us) +1750 =43429432	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2000	MAC payload L	LQI 123	FCS OK
P.nbr. RX 14	Time (us) +83461 =43512893	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4000	MAC payload L	LQI 171	FCS OK
P.nbr. RX 15	Time (us) +2121 =43515014	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x04	Dest. PAN 0xCAFE	Dest. Address 0x2A67	Source Address 0x0000	MAC payload 3 *	LQI 120	FCS OK
P.nbr. RX 16	Time (us) +86135 =43601149	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x6000	MAC payload L	LQI 118	FCS OK

Figura 3.16. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura,

se observa de acuerdo con la Figura 3.17 el nodo de tercer nivel con la dirección 0x0D2E le responde con el mensaje “NEED_ID_L” al nodo asignador con la dirección 0x4000, luego el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x6494 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x0D2E define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “L”, seguidamente el nodo de tercer nivel con la dirección 0x4E10 le responde con el mensaje “NEED_ID_L” al nodo asignador con la dirección 0x4800, luego el nodo asignador con la dirección 0x4800 envía información para que el nodo con dirección 0x4E10 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x4E10 define su nueva dirección con el valor 0x4A00 y envía una trama en broadcast con el mensaje “L”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3053821		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L	136	OK
17	=46654970	20	DATA	0	0	0	1	0x01	0xCAFE	0x4000	0x0D2E		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3160326		Type	Sec	Pnd	Ack.req	PAN_compr				1	171	OK
18	=49815296	13	DATA	0	0	0	1	0x03	0xCAFE	0x0D2E	0x4000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+86598		Type	Sec	Pnd	Ack.req	PAN_compr				L	139	OK
19	=49901894	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4800		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3087197		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L	99	OK
20	=52989091	20	DATA	0	0	0	1	0x01	0xCAFE	0x4800	0x4E10		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3227525		Type	Sec	Pnd	Ack.req	PAN_compr				1	139	OK
21	=56216616	13	DATA	0	0	0	1	0x03	0xCAFE	0x4E10	0x4800	-	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84784		Type	Sec	Pnd	Ack.req	PAN_compr				L	99	OK
22	=56301400	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4A00		

Figura 3.17. Asignación de direcciones de los nodos asignadores.

Para probar el escenario de nodo caído se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.19, por lo tanto, una vez que el proceso de asignación de direcciones finalizó, al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.18.

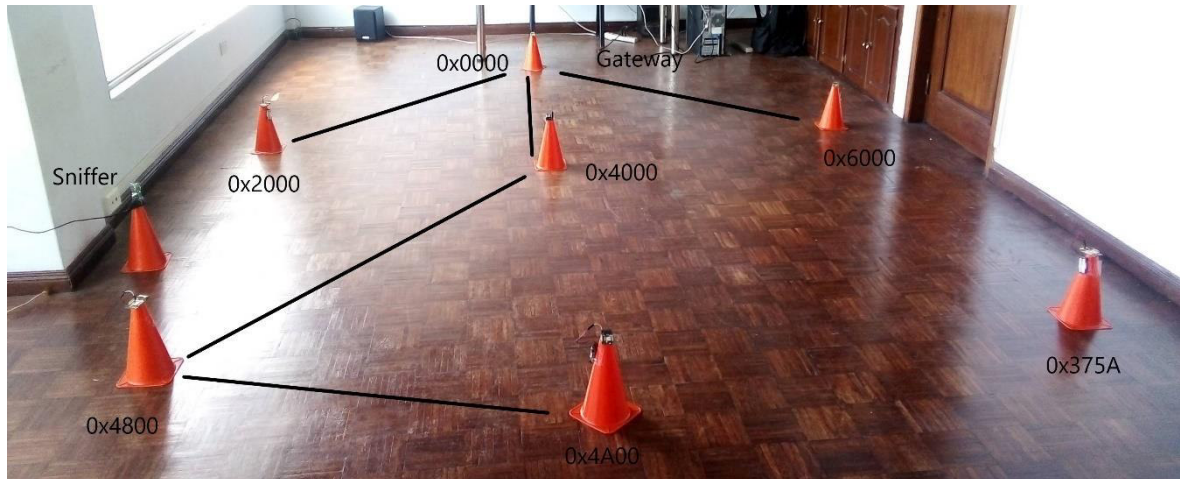


Figura 3.18. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+29330374	16	Type	Sec	Pnd	Ack.req	PAN_compr				MY_ID		
23	=85631774	16	DATA	0	0	0	1	0x00	0xCAFE	0x375A	0x375A	88	OK

Figura 3.19. Dirección aleatoria del nodo que se encendió.

Una vez que el nodo se enciende y define su dirección aleatoria con el valor de 0x375A, envía una trama broadcast con el mensaje “CAIDO”, los nodos asignadores que se encuentren dentro de la zona de cobertura del nodo le responden con una trama unicast con el mensaje “L”, en este caso el nodo asignador con la dirección 0x4A00 le respondió al nodo con la dirección 0x375A, luego el nodo con la dirección 0x375A le responde al nodo asignador con el mensaje “NEED_ID_L”, hecho esto el nodo asignador le envía al nodo la información para que calcule y defina su nueva dirección, luego de esto el nodo define su nueva dirección con el valor de 0x4A80 se convierte en nodo asignador y envía una trama broadcast con el mensaje “L”, como se observa en la Figura 3.20.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4312876	16	Type	Sec	Pnd	Ack.req	PAN_compr				CAI		
24	=89944650	16	DATA	0	0	0	1	0x01	0xCAFE	0xFFFF	0x375A	72	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3622	12	Type	Sec	Pnd	Ack.req	PAN_compr				L		
25	=89948272	12	DATA	0	0	0	1	0x03	0xCAFE	0x375A	0x4A00	93	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3177143	20	Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_L		
26	=93125415	20	DATA	0	0	0	1	0x02	0xCAFE	0x4A00	0x375A	91	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3172290	13	Type	Sec	Pnd	Ack.req	PAN_compr				1		
27	=96297705	13	DATA	0	0	0	1	0x04	0xCAFE	0x375A	0x4A00	99	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+85116	12	Type	Sec	Pnd	Ack.req	PAN_compr				L		
28	=96382821	12	DATA	0	0	0	1	0x03	0xCAFE	0xFFFF	0x4A80	91	OK

Figura 3.20. Asignación de dirección al nodo caído.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.21, además una vez finalizado este proceso de acuerdo con la Figura 3.20 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 96.38 segundos.

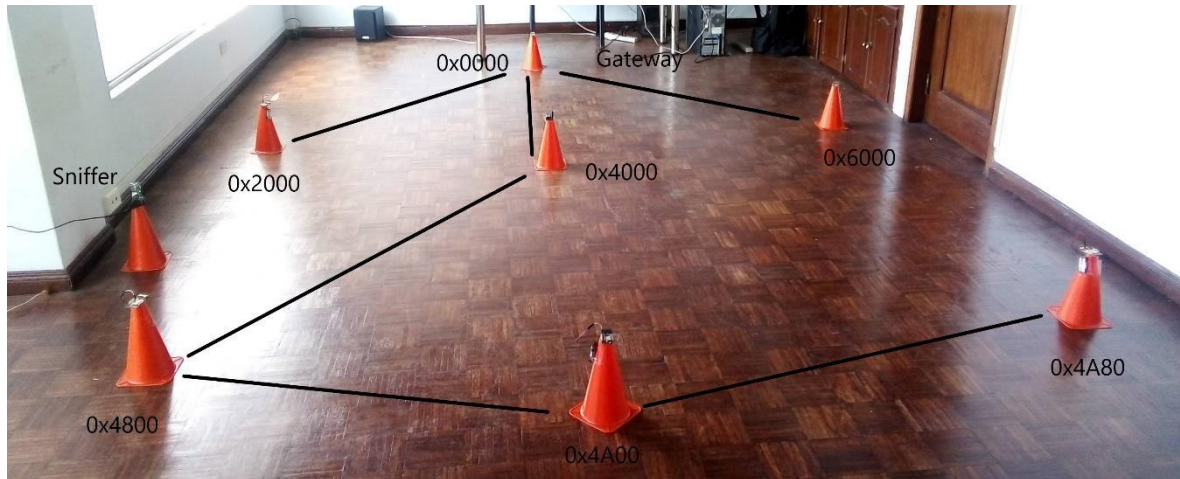


Figura 3.21. Topología final.

3.4. PRUEBA 4

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.22, cabe mencionar que la distancia entre el Gateway y el primer nivel es de 1.75 m, la distancia entre los nodos del primer nivel es 1.5 m, la distancia entre los nodos del segundo y tercer nivel es 2 m y por último la distancia entre los nodos del tercer nivel es 1 m, en esta prueba se presenta un escenario sin fallos, el criterio para el proceso de asignación de direcciones será por nivel de potencia de la trama recibida, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.23.



Figura 3.22. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
1	=0	16	DATA	0	0	0	1	0x00	0xCAFE	0x28EC	0x28EC	107	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3883442		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
2	=3883442	16	DATA	0	0	0	1	0x00	0xCAFE	0x78E9	0x78E9	120	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5486466		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
3	=9369908	16	DATA	0	0	0	1	0x00	0xCAFE	0x109E	0x109E	99	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+7531192		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
4	=16901100	16	DATA	0	0	0	1	0x00	0xCAFE	0x4822	0x4822	128	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3988321		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
5	=20889421	16	DATA	0	0	0	1	0x00	0xCAFE	0x76D7	0x76D7	128	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3059960		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
6	=23949381	16	DATA	0	0	0	1	0x00	0xCAFE	0x392E	0x392E	102	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3838731		Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
7	=27788112	16	DATA	0	0	0	1	0x00	0xCAFE	0x538E	0x538E	69	OK

Figura 3.23. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “P”, luego los dos nodos correspondientes al segundo nivel responden al Gateway con el mensaje “NEED_ID_P”, en el siguiente orden primero responde el nodo con la dirección 0x109E, luego el nodo con la dirección 0x78E9. De acuerdo con el orden de las tramas de la Figura 3.24, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que defina su dirección fija el nodo con la dirección 0x109E, después envía la información para que defina su dirección fija el nodo con la dirección 0x78E9, dado que el nodo con la dirección 0x109E calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “P”, luego el nodo con la dirección 0x78E9 define su nueva dirección con el valor 0x4000 y envía una trama en broadcast con el mensaje “P”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+9562163		Type	Sec	Pnd	Ack.req	PAN_compr				P	75	OK
8	=37350275	12	DATA	0	0	0	1	0x01	0xCAFE	0xFFFF	0x0000		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3075009		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	161	OK
9	=40425284	20	DATA	0	0	0	1	0x01	0xCAFE	0x0000	0x109E		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+11057		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	139	OK
10	=40436341	20	DATA	0	0	0	1	0x01	0xCAFE	0x0000	0x78E9		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3335218		Type	Sec	Pnd	Ack.req	PAN_compr				1	96	OK
11	=43771559	13	DATA	0	0	0	1	0x02	0xCAFE	0x109E	0x0000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84927		Type	Sec	Pnd	Ack.req	PAN_compr				2	96	OK
12	=43856486	13	DATA	0	0	0	1	0x03	0xCAFE	0x78E9	0x0000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+842		Type	Sec	Pnd	Ack.req	PAN_compr				P	161	OK
13	=43857328	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2000		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+85572		Type	Sec	Pnd	Ack.req	PAN_compr				P	139	OK
14	=43942900	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4000		

Figura 3.24. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.25 que los nodos del tercer nivel con las direcciones 0x76D7, 0x392E y 0x538E respondieron con el mensaje "NEED_ID_P" al nodo asignador con la dirección 0x2000, luego el nodo con la dirección 0x4822 le responde con el mensaje "NEED_ID_P" al nodo asignador con la dirección 0x4000, después el nodo asignador con la dirección 0x2000 envía información para que los nodos con dirección 0x76D7 y 0x392E calculen y definan su nueva dirección, hecho esto el nodo con dirección 0x76D7 define su nueva dirección con el valor 0x2800 y envía una trama en broadcast con el mensaje "P", luego el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x4822 calcule y defina su nueva dirección, también el nodo asignador con la dirección 0x2000 envía información para que el nodo restante con dirección 0x538E calcule y defina su nueva dirección, seguidamente el nodo con dirección 0x382E define su nueva dirección con el valor 0x3000 y envía una trama en broadcast con el mensaje "P", después el nodo con dirección 0x4822 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje "P", finalmente el nodo con dirección 0x538E define su nueva dirección con el valor 0x3800 y envía una trama en broadcast con el mensaje "P".

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+2977578		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	136	OK
15	=46920478	20	DATA	0	0	0	1	0x01	0xCAFE	0x2000	0x76D7		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+1382		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	123	OK
16	=46921860	20	DATA	0	0	0	1	0x01	0xCAFE	0x2000	0x392E		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+34499		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	102	OK
17	=46956359	20	DATA	0	0	0	1	0x01	0xCAFE	0x2000	0x538E		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+47307		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P	110	OK
18	=47003666	20	DATA	0	0	0	1	0x01	0xCAFE	0x4000	0x4822		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3075820		Type	Sec	Pnd	Ack.req	PAN_compr				1	161	OK
19	=50079486	13	DATA	0	0	0	1	0x03	0xCAFE	0x76D7	0x2000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84390		Type	Sec	Pnd	Ack.req	PAN_compr				2	161	OK
20	=50163876	13	DATA	0	0	0	1	0x04	0xCAFE	0x392E	0x2000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+2760		Type	Sec	Pnd	Ack.req	PAN_compr				p	134	OK
21	=50166636	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2800		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+15039		Type	Sec	Pnd	Ack.req	PAN_compr				1	139	OK
22	=50181675	13	DATA	0	0	0	1	0x03	0xCAFE	0x4822	0x4000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+66307		Type	Sec	Pnd	Ack.req	PAN_compr				3	161	OK
23	=50247982	13	DATA	0	0	0	1	0x05	0xCAFE	0x538E	0x2000	*	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+1448		Type	Sec	Pnd	Ack.req	PAN_compr				p	120	OK
24	=50249430	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x3000		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+19924		Type	Sec	Pnd	Ack.req	PAN_compr				p	110	OK
25	=50269354	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4800		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+66176		Type	Sec	Pnd	Ack.req	PAN_compr				p	102	OK
26	=50335530	12	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x3800		

Figura 3.25. Asignación de direcciones de los nodos asignadores.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.26, además una vez finalizado este proceso de acuerdo con la Figura 3.25 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 50.34 segundos.

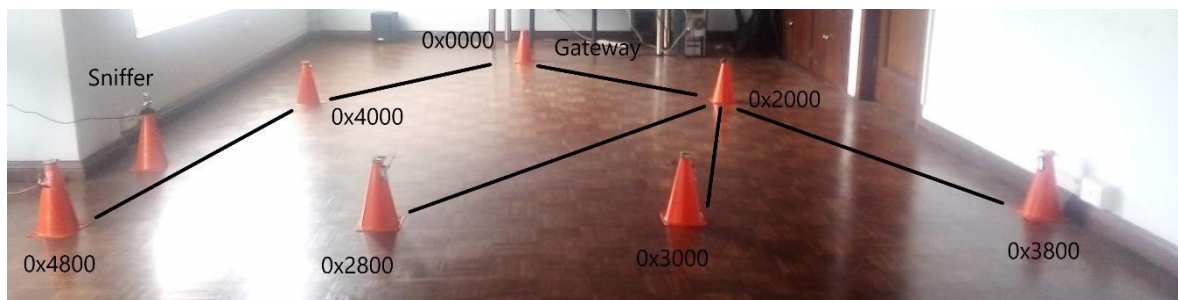


Figura 3.26. Topología final.

3.5. PRUEBA 5

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.27, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 4, en esta prueba se presenta un escenario con enlace ruidoso, el criterio para el proceso de asignación de direcciones será por nivel de potencia de la trama recibida, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.28. Dado que es un escenario con enlace ruidoso se debe mencionar que el nodo configurado para simular dicho escenario es el que tiene la dirección aleatoria 0x547D, como se explicó en Código 2.11 en el nodo mencionado la variable “enlaceRuidoso” tiene el valor de *true*.



Figura 3.27. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
1	=0		DATA	0	0	0	1	0x00	0xCAFE	0x7015	0x7015	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4122072	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
2	=4122072		DATA	0	0	0	1	0x00	0xCAFE	0x77EC	0x77EC	118	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6883806	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
3	=11005878		DATA	0	0	0	1	0x00	0xCAFE	0x5FD7	0x5FD7	75	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6734606	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
4	=17740484		DATA	0	0	0	1	0x00	0xCAFE	0x5DD1	0x5DD1	155	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3792325	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
5	=21532809		DATA	0	0	0	1	0x00	0xCAFE	0x547D	0x547D	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+9251774	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
6	=30784583		DATA	0	0	0	1	0x00	0xCAFE	0x767D	0x767D	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3389595	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
7	=34174178		DATA	0	0	0	1	0x00	0xCAFE	0x1B75	0x1B75	83	OK

Figura 3.28. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “P”, los dos nodos correspondientes al segundo nivel le responden con el mensaje “NEED_ID_P”, en

el siguiente orden primero responde el nodo con la dirección 0x5FD7, luego el nodo con la dirección 0x77EC. De acuerdo con el orden de las tramas de la Figura 3.29, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que defina su dirección fija el nodo con la dirección 0x5FD7, dado que el nodo con la dirección 0x5FD7 calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “P”, seguidamente el Gateway envía la información para que defina su dirección fija el nodo con la dirección 0x77EC, dado que el nodo con la dirección 0x77EC calcula y define su nueva dirección con el valor de 0x4000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “P”, luego como se mencionó en la sección 2.1.1.1 en el caso de enlaces ruidosos el nodo configurado en este escenario, es decir, una vez que los nodos asignadores dentro de la zona de cobertura del nodo con la dirección 0x547D envían una trama broadcast con el mensaje “P”, el nodo con el que se simula el enlace ruidoso no procesa la primera trama y espera 1 segundo para enviar una trama unicast con el mensaje “FALLO” al nodo asignador con la dirección 0x2000, como se observa en la trama 15 de la Figura 3.29 el nodo envía la trama después de alrededor de un segundo y medio, después el nodo asignador con la dirección 0x2000 responde al nodo con una trama unicast y el mensaje “P”.

P.nbr. RX 8	Time (us) +9460965 =43635143	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x0000	MAC payload P	LQI 110	FCS OK
P.nbr. RX 9	Time (us) +3074586 =46709729	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x5FD7	MAC payload NEED_ ID_P	LQI 161	FCS OK
P.nbr. RX 10	Time (us) +10792 =46720521	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x77EC	MAC payload NEED_ ID_P	LQI 139	FCS OK
P.nbr. RX 11	Time (us) +3335306 =50055827	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x5FD7	Source Address 0x0000	MAC payload 1 *	LQI 99	FCS OK
P.nbr. RX 12	Time (us) +85097 =50140924	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2000	MAC payload P	LQI 161	FCS OK
P.nbr. RX 13	Time (us) +4103 =50145027	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x77EC	Source Address 0x0000	MAC payload 2 *	LQI 96	FCS OK
P.nbr. RX 14	Time (us) +86319 =50231346	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4000	MAC payload P	LQI 139	FCS OK
P.nbr. RX 15	Time (us) +1570533 =51801879	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x2000	Source Address 0x547D	MAC payload FAL LO	LQI 136	FCS OK
P.nbr. RX 16	Time (us) +3610 =51805489	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x547D	Source Address 0x2000	MAC payload P	LQI 161	FCS OK

Figura 3.29. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.30 los nodos de tercer nivel con las direcciones 0x767D, 0x1B75 y 0x547D le responden con el mensaje "NEED_ID_P" al nodo asignador con la dirección 0x2000, luego el nodo asignador con la dirección 0x2000 envía información para que los nodos con dirección 0x767D y 0x1B75 calculen y definan su nueva dirección, hecho esto el nodo con dirección 0x767D define su nueva dirección con el valor 0x2800 y envía una trama en broadcast con el mensaje "P", seguidamente el nodo asignador con la dirección 0x2000 envía información para que el nodo con dirección 0x547D calcule y defina su nueva dirección, después el nodo con dirección 0x1B75 define su nueva dirección con el valor 0x3000 y envía una trama en broadcast con el mensaje "P", posteriormente el nodo con dirección 0x547D define su nueva dirección con el valor 0x3800 y envía una trama en broadcast con el mensaje "P", luego el nodo con la dirección 0x5DD1 le responde con un mensaje "NEED_ID_P" al nodo asignador con la dirección 0x2800, después dicho nodo asignador envía información para que el nodo con dirección 0x5DD1 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x5DD1 define su nueva dirección con el valor 0x2A00 y envía una trama en broadcast con el mensaje "P".

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1401696		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P			
17	=53207185	20	DATA	0	0	0	1	0x01	0xC9FE	0x2000	0x767D	123	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+33786		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P			
18	=53240971	20	DATA	0	0	0	1	0x01	0xC9FE	0x2000	0x1B75	102	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1704864		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P			
19	=54945835	20	DATA	0	0	0	1	0x02	0xC9FE	0x2000	0x547D	136	OK	
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1421304		Type	Sec	Pnd	Ack.req	PAN_compr				1			
20	=56367139	13	DATA	0	0	0	1	0x04	0xC9FE	0x767D	0x2000	*	161	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+81788		Type	Sec	Pnd	Ack.req	PAN_compr				2			
21	=56448927	13	DATA	0	0	0	1	0x05	0xC9FE	0x1B75	0x2000	*	161	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+2455		Type	Sec	Pnd	Ack.req	PAN_compr				P			
22	=56451382	12	DATA	0	0	0	1	0x02	0xC9FE	0xFFFF	0x2800		126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+81317		Type	Sec	Pnd	Ack.req	PAN_compr				3			
23	=56532699	13	DATA	0	0	0	1	0x06	0xC9FE	0x547D	0x2000	*	161	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+2084		Type	Sec	Pnd	Ack.req	PAN_compr				P			
24	=56534783	12	DATA	0	0	0	1	0x02	0xC9FE	0xFFFF	0x3000		102	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+84377		Type	Sec	Pnd	Ack.req	PAN_compr				P			
25	=56619160	12	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	0x3800		136	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+2972940		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_ID_P			
26	=59592100	20	DATA	0	0	0	1	0x01	0xC9FE	0x2800	0x5DD1		139	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+3152749		Type	Sec	Pnd	Ack.req	PAN_compr				1			
27	=62744849	13	DATA	0	0	0	1	0x03	0xC9FE	0x5DD1	0x2800	-	131	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+88489		Type	Sec	Pnd	Ack.req	PAN_compr				P			
28	=62833338	12	DATA	0	0	0	1	0x02	0xC9FE	0xFFFF	0x2A00		142	OK

Figura 3.30. Asignación de direcciones de los nodos asignadores.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.31, además una vez finalizado este proceso de acuerdo con la Figura 3.30 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 62.83 segundos.

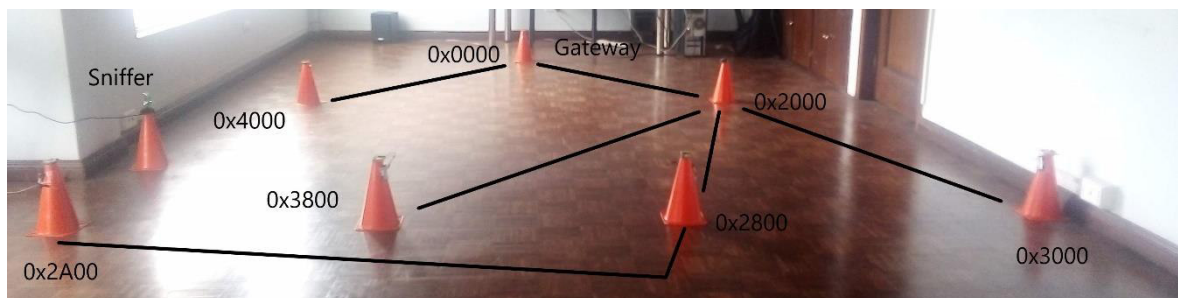


Figura 3.31. Topología final.

3.6. PRUEBA 6

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.32, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 4, en esta prueba se presenta un escenario con nodo caído, el criterio para el proceso de asignación de direcciones será por nivel de potencia de la trama recibida, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.33. Dado que es un escenario con nodo caído se debe mencionar que el nodo configurado para simular dicho escenario es el que no tiene dirección aleatoria en la Figura 3.32 ya que al inicio se encuentra apagado y una vez finalizado el proceso de asignación de direcciones se encendió manualmente para probar el escenario con nodo caído, como se explicó en Código 2.11 en el nodo mencionado la variable “nodoCaído” tiene el valor de *true*.



Figura 3.32. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr. RX 1	Time (us) +0 =0	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x3B5C	Source Address 0x3B5C	MAC payload MY_ ID	LQI 118	FCS OK
P.nbr. RX 2	Time (us) +4796728 =4796728	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x775C	Source Address 0x775C	MAC payload MY_ ID	LQI 126	FCS OK
P.nbr. RX 3	Time (us) +7348640 =12145368	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x5F6D	Source Address 0x5F6D	MAC payload MY_ ID	LQI 72	FCS OK
P.nbr. RX 4	Time (us) +8930900 =21076268	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x732F	Source Address 0x732F	MAC payload MY_ ID	LQI 163	FCS OK
P.nbr. RX 5	Time (us) +5835250 =26911518	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x21C7	Source Address 0x21C7	MAC payload MY_ ID	LQI 134	FCS OK
P.nbr. RX 6	Time (us) +10355901 =37267419	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x00	Dest. PAN 0xCAFE	Dest. Address 0x11E8	Source Address 0x11E8	MAC payload MY_ ID	LQI 139	FCS OK

Figura 3.33. Direcciones aleatorias asignadas a los nodos.

Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “P”, luego los dos nodos correspondientes al segundo nivel responden al Gateway con el mensaje

“NEED_ID_P”, en el siguiente orden primero responde el nodo con la dirección 0x5F6D y luego el nodo con la dirección 0x775C. De acuerdo con el orden de las tramas de la Figura 3.34, el Gateway respondió según el orden en que los nodos le respondieron, envía la información para que definan su dirección fija los nodos con la dirección 0x5F6D y 0x775C respectivamente, dado que el nodo con la dirección 0x5F6D calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “P”, seguidamente el nodo con la dirección 0x775C calcula y define su nueva dirección con el valor de 0x4000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “P”.

P.nbr. RX 7	Time (us) +14583612 =51851031	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x0000	MAC payload P	LQI 102	FCS OK
P.nbr. RX 8	Time (us) +3075311 =54926342	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x5F6D	MAC payload NEED_ ID_P	LQI 161	FCS OK
P.nbr. RX 9	Time (us) +7528 =54933870	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x775C	MAC payload NEED_ ID_P	LQI 131	FCS OK
P.nbr. RX 10	Time (us) +3337635 =58271505	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x5F6D	Source Address 0x0000	MAC payload 1 *	LQI 96	FCS OK
P.nbr. RX 11	Time (us) +84898 =58356403	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x775C	Source Address 0x0000	MAC payload 2 *	LQI 93	FCS OK
P.nbr. RX 12	Time (us) +2245 =58358648	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2000	MAC payload P	LQI 163	FCS OK
P.nbr. RX 13	Time (us) +83484 =58442132	Length 12	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4000	MAC payload P	LQI 128	FCS OK

Figura 3.34. Asignación de direcciones del Gateway.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.35 los nodos de tercer nivel con dirección 0x21C7 y 0x11E8 le responden con el mensaje “NEED_ID_P” al nodo asignador con la dirección 0x2000, mientras que el nodo de tercer nivel con la dirección 0x732F le responde con el mensaje “NEED_ID_P” al nodo asignador con la dirección 0x4000, luego el nodo asignador con la dirección 0x2000 envía la información para que definan su dirección fija los nodos con la dirección 0x21C7 y 0x11E8 respectivamente, hecho esto el nodo con dirección 0x21C7 define su nueva dirección con el valor 0x2800 y envía una trama en broadcast con el mensaje “P”, seguidamente el nodo con dirección 0x11E8 define su nueva dirección con el valor 0x3000 y envía una trama en broadcast con el mensaje “P”, luego el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x732F calcule y defina su nueva dirección, hecho esto el nodo con dirección

0x732F define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “P”.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 14	+2979602 =61421734	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x2000	0x21C7	NEED_ID_P	139	OK
RX 15	+3305 =61425039	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x2000	0x11E8	NEED_ID_P	128	OK
RX 16	+158224 =61583263	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4000	0x732F	NEED_ID_P	150	OK
RX 17	+2996295 =64579558	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x21C7	0x2000	1 *	163	OK
RX 18	+83090 =64662648	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x11E8	0x2000	2 *	163	OK
RX 19	+2303 =64664951	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x2800	P	139	OK
RX 20	+83873 =64748824	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x3000	P	131	OK
RX 21	+3182 =64752006	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x732F	0x4000	1 *	128	OK
RX 22	+86911 =64838917	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4800	P	153	OK

Figura 3.35. Asignación de direcciones de los nodos asignadores.

Para probar el escenario de nodo caído se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.37, por lo tanto, una vez que el proceso de asignación de direcciones finalizó, al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.36.

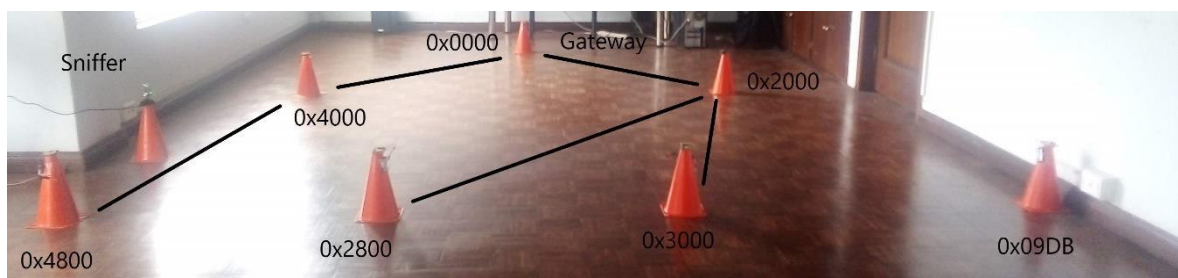


Figura 3.36. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 23	+40410937 =105249854	16	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x00	0xCAFE	0x09DB	0x09DB	MY_ID	102	OK

Figura 3.37. Dirección aleatoria del nodo que se encendió.

Una vez que el nodo se enciende y define su dirección aleatoria con el valor de 0x09DB, envía una trama broadcast con el mensaje “CAIDO”, los nodos asignadores que se encuentren dentro de la zona de cobertura del nodo le responden con una trama unicast con el mensaje “P”, en este caso el nodo asignador con la dirección 0x3000 le respondió al nodo con la dirección 0x09DB, como se puede ver de acuerdo al criterio de nivel de potencia de recibida le responde el nodo asignador que se encuentra más cerca del nodo caído, esto debido a que al estar más cerca la trama recibida tendrá mayor potencia, luego el nodo con la dirección 0x09DB le responde al nodo asignador con el mensaje “NEED_ID_P”, hecho esto el nodo asignador le envía al nodo la información para que calcule y defina su nueva dirección, luego de esto el nodo define su nueva dirección con el valor de 0x3200 se convierte en nodo asignador y envía una trama broadcast con el mensaje “P”, como se observa en la Figura 3.38.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6593251		Type	Sec	Pnd	Ack.req	PAN_compr				CAI		
24	=111843105	16	DATA	0	0	0	1	0x01	0xCAFE	0xFFFF	DO	102	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+2025		Type	Sec	Pnd	Ack.req	PAN_compr				P		
25	=111845130	12	DATA	0	0	0	1	0x03	0xCAFE	0x09DB		128	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3179932		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_		
26	=115025062	20	DATA	0	0	0	1	0x02	0xCAFE	0x3000	ID_P	88	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3148925		Type	Sec	Pnd	Ack.req	PAN_compr				1		
27	=118173987	13	DATA	0	0	0	1	0x04	0xCAFE	0x09DB	-	131	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+85425		Type	Sec	Pnd	Ack.req	PAN_compr				P		
28	=118259412	12	DATA	0	0	0	1	0x03	0xCAFE	0xFFFF		85	OK

Figura 3.38. Asignación de dirección al nodo caído.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.39, además una vez finalizado este proceso de acuerdo con la Figura 3.38 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 118.26 segundos.

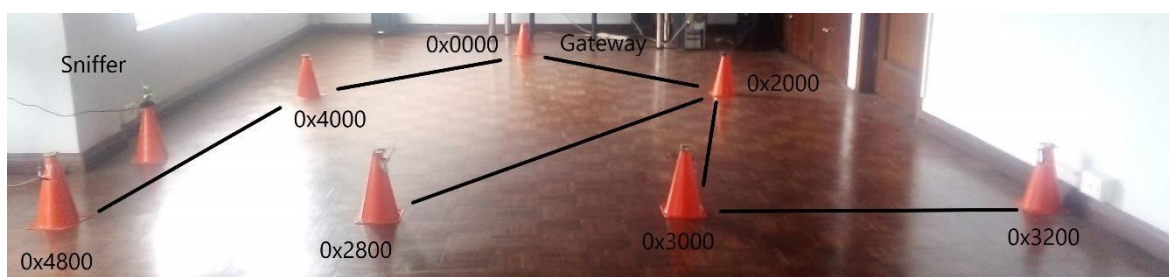


Figura 3.39. Topología final.

3.7. PRUEBA 7

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.40, cabe mencionar que la distancia entre el Gateway y el primer nivel es de 1.75 m, la distancia entre los nodos del primer nivel es 1.5 m, la distancia entre los nodos del segundo y tercer nivel es 2 m, la distancia entre los nodos del tercer nivel es 1 m y por último la distancia entre los nodos del tercer y cuarto nivel es de 2 m, en esta prueba se presenta un escenario sin fallos, el criterio para el proceso de asignación de direcciones será por número de saltos, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.41.



Figura 3.40. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
1	=0		DATA	0	0	0	1	0x00	0xC9FE	0x452F	0x452F	67	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3894530	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
2	=3894530		DATA	0	0	0	1	0x00	0xC9FE	0x7228	0x7228	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5435597	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
3	=9330127		DATA	0	0	0	1	0x00	0xC9FE	0x44EB	0x44EB	110	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5705318	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
4	=15035445		DATA	0	0	0	1	0x00	0xC9FE	0x2CBE	0x2CBE	163	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3844415	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
5	=18879860		DATA	0	0	0	1	0x00	0xC9FE	0x3F34	0x3F34	88	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4835043	16	Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
6	=23714903		DATA	0	0	0	1	0x00	0xC9FE	0x709D	0x709D	120	OK

Figura 3.41. Direcciones aleatorias asignadas a los nodos.

En la Figura 3.42 se observa que el mensaje “S” está acompañado de un asterisco, esto es debido a que se envió como valor hexadecimal en lugar de transformarlo a texto y después enviarlo, por lo que se hará uso de la Figura 3.43 para poder visualizar el valor de dicho asterisco que corresponde al número de saltos. Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “S01”, luego los dos nodos correspondientes al segundo nivel responden al Gateway con el mensaje “NEED_ID_S”, en el siguiente orden primero responde el nodo con la dirección 0x44EB, luego el nodo con la dirección 0x7228. De acuerdo con el orden de las tramas de la Figura 3.42, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que defina su dirección fija el nodo con la dirección 0x44EB, después envía la información para que defina su dirección fija el nodo con la dirección 0x7228, dado que el nodo con la dirección 0x44EB calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “S02”, luego el nodo con la dirección 0x7228 define su nueva dirección con el valor 0x4000 y envía una trama en broadcast con el mensaje “S02”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+9817122		Type	Sec	Pnd	Ack.req	PAN_compr			S				
7	=33532025	13	DATA	0	0	0	1	0x01	0xCAFE	0xFFFF	0x0000	*	118	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+3074628		Type	Sec	Pnd	Ack.req	PAN_compr			NEED_ID_S				
8	=36606653	20	DATA	0	0	0	1	0x01	0xCAFE	0x0000	0x44EB	*	155	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+8038		Type	Sec	Pnd	Ack.req	PAN_compr			NEED_ID_S				
9	=36614691	20	DATA	0	0	0	1	0x01	0xCAFE	0x0000	0x7228	*	131	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+3335925		Type	Sec	Pnd	Ack.req	PAN_compr			1				
10	=39950616	13	DATA	0	0	0	1	0x02	0xCAFE	0x44EB	0x0000	*	72	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+84521		Type	Sec	Pnd	Ack.req	PAN_compr			2				
11	=40035137	13	DATA	0	0	0	1	0x03	0xCAFE	0x7228	0x0000	*	72	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1232		Type	Sec	Pnd	Ack.req	PAN_compr			S				
12	=40036369	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2000	*	155	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+85766		Type	Sec	Pnd	Ack.req	PAN_compr			S				
13	=40122135	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4000	*	131	OK

Figura 3.42. Asignación de direcciones del Gateway.

Como se puede observar en la Figura 3.43 el valor 53 en texto equivale al carácter “S” en texto, además se sabe que el mensaje “S” va acompañado del contador de saltos por lo que el Gateway envía en broadcast un mensaje “S01”, mientras que los nodos asignadores envían en broadcast un mensaje “S02”, según esto se tiene que debido a que los nodos que se convirtieron en nodos asignadores primero recibieron y extrajeron el valor del contador 1, al convertirse en nodos asignadores aumentan el contador a 2 y así seguirá ocurriendo sucesivamente cumpliendo con lo mencionado en la sección 2.1.1.3 y 2.2.3. Según esto el valor que los nodos sin dirección fija extraen del contador corresponde al número de saltos que tendrían que dar para llegar al Gateway.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+9817122		Type	Sec	Pnd	Ack.req	PAN_compr			53				
7	=33532025	13	DATA	0	0	0	1	0x01	0xCAFE	0xFFFF	0x0000	01	118	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+1232		Type	Sec	Pnd	Ack.req	PAN_compr			53				
12	=40036369	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2000	02	155	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+85766		Type	Sec	Pnd	Ack.req	PAN_compr			53				
13	=40122135	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4000	02	131	OK

Figura 3.43. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.44 que el nodo de tercer nivel con la dirección 0x3F34 responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x2000, después el nodo de tercer nivel con la dirección 0x2CBE responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x4000, posteriormente el

nodo asignador con la dirección 0x2000 envía información para que el nodo con dirección 0x3F34 calcule y defina su nueva dirección, también el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x2CBE calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x3F34 define su nueva dirección con el valor 0x2800 y envía una trama en broadcast con el mensaje “S03”, seguidamente el nodo con dirección 0x2CBE define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “S03”, después el nodo de cuarto nivel con la dirección 0x709D responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x2800, posteriormente el nodo asignador con la dirección 0x2800 envía información para que el nodo con dirección 0x709D calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x709D define su nueva dirección con el valor 0x2A00 y envía una trama en broadcast con el mensaje “S04”. Aquí también se observa que cada vez que un nodo se convierte en nodo asignador cuando envía la trama broadcast el contador que acompaña a la letra “S” aumenta, como se explicó en Figura 3.43, esto se puede observar en la Figura 3.45.

P.nbr. RX 14	Time (us) +3014513 =43136648	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x2000	Source Address 0x3F34	MAC payload NEED_ID_S	LQI 107	FCS OK
P.nbr. RX 15	Time (us) +49179 =43185827	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x4000	Source Address 0x2CBE	MAC payload NEED_ID_S	LQI 158	FCS OK
P.nbr. RX 16	Time (us) +3111677 =46297504	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x3F34	Source Address 0x2000	MAC payload 1 *	LQI 155	FCS OK
P.nbr. RX 17	Time (us) +55846 =46353350	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x2CBE	Source Address 0x4000	MAC payload 1 *	LQI 131	FCS OK
P.nbr. RX 18	Time (us) +30943 =46384293	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2800	MAC payload S *	LQI 104	FCS OK
P.nbr. RX 19	Time (us) +53284 =46437577	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4800	MAC payload S *	LQI 158	FCS OK
P.nbr. RX 20	Time (us) +3091973 =49529550	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x2800	Source Address 0x709D	MAC payload NEED_ID_S	LQI 88	FCS OK
P.nbr. RX 21	Time (us) +3185480 =52715030	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x709D	Source Address 0x2800	MAC payload 1 -	LQI 107	FCS OK
P.nbr. RX 22	Time (us) +86689 =52801719	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2A00	MAC payload S *	LQI 88	FCS OK

Figura 3.44. Asignación de direcciones de los nodos asignadores.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+30943		Type	Sec	Pnd	Ack.req	PAN_compr				53			
18	=46384293	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2800	03	104	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+53284		Type	Sec	Pnd	Ack.req	PAN_compr				53			
19	=46437577	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x4800	03	158	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS	
RX	+86689		Type	Sec	Pnd	Ack.req	PAN_compr				53			
22	=52801719	13	DATA	0	0	0	1	0x02	0xCAFE	0xFFFF	0x2A00	04	88	OK

Figura 3.45. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

En este caso el escenario para probar el algoritmo es similar al escenario de nodo caído explicado en los anteriores criterios, es decir, se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.47, por lo tanto, una vez que el proceso de asignación de direcciones finalizó al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.48.

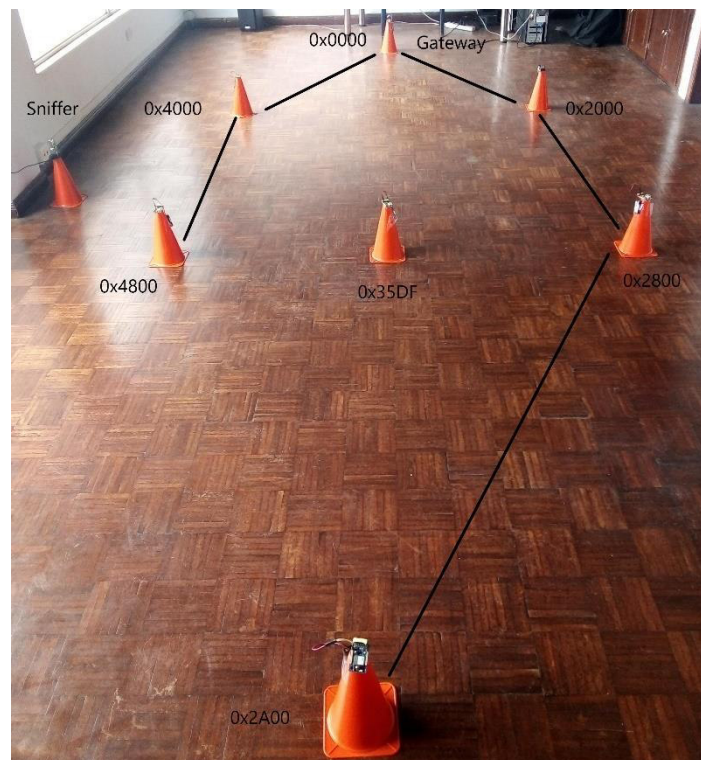


Figura 3.46. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+23420402		Type	Sec	Pnd	Ack.req	PAN_compr				MY_ID		
23	=76222121	16	DATA	0	0	0	1	0x00	0xCAFE	0x35DF	0x35DF	134	OK

Figura 3.47. Dirección aleatoria del nodo que se encendió.

Cuando el nodo se enciende, define su dirección aleatoria con el valor de 0x35DF, aunque el proceso es parecido al del escenario de nodo caído en este caso el nodo envía

una trama broadcast con el mensaje “SALTO”, los nodos asignadores que se encuentren dentro de la zona de cobertura del nodo le responden con una trama unicast, en este escenario el nodo responderá la oferta del nodo asignador por el que menos número de saltos para llegar al Gateway tenga que pasar, en este caso el nodo asignador con la dirección 0x4800 le respondió con el mensaje “S03” al nodo con la dirección 0x35DF, luego el nodo con la dirección 0x35DF le responde al nodo asignador con el mensaje “NEED_ID_S”, hecho esto el nodo asignador le envía al nodo la información para que calcule y defina su nueva dirección, luego de esto el nodo define su nueva dirección con el valor de 0x4A00 se convierte en nodo asignador y envía una trama broadcast con el mensaje “S04”.

P.nbr. RX 24	Time (us) +3527048 =79749169	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x35DF	MAC payload SAL TO	LQI 120	FCS OK
P.nbr. RX 25	Time (us) +1385 =79750554	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x35DF	Source Address 0x4800	MAC payload S *	LQI 155	FCS OK
P.nbr. RX 26	Time (us) +3148065 =82898619	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x4800	Source Address 0x35DF	MAC payload NEED_ ID_S	LQI 102	FCS OK
P.nbr. RX 27	Time (us) +3147294 =86045913	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x04	Dest. PAN 0xCAFE	Dest. Address 0x35DF	Source Address 0x4800	MAC payload 1 -	LQI 158	FCS OK
P.nbr. RX 28	Time (us) +84868 =86130781	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4A00	MAC payload S *	LQI 107	FCS OK

Figura 3.48. Asignación de dirección al nodo para probar escenario.

P.nbr. RX 25	Time (us) +1385 =79750554	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x35DF	Source Address 0x4800	MAC payload 53 03	LQI 155	FCS OK
P.nbr. RX 28	Time (us) +84868 =86130781	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4A00	MAC payload 53 04	LQI 107	FCS OK

Figura 3.49. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.50, además una vez finalizado este proceso de acuerdo con la Figura 3.48 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 86.13 segundos.

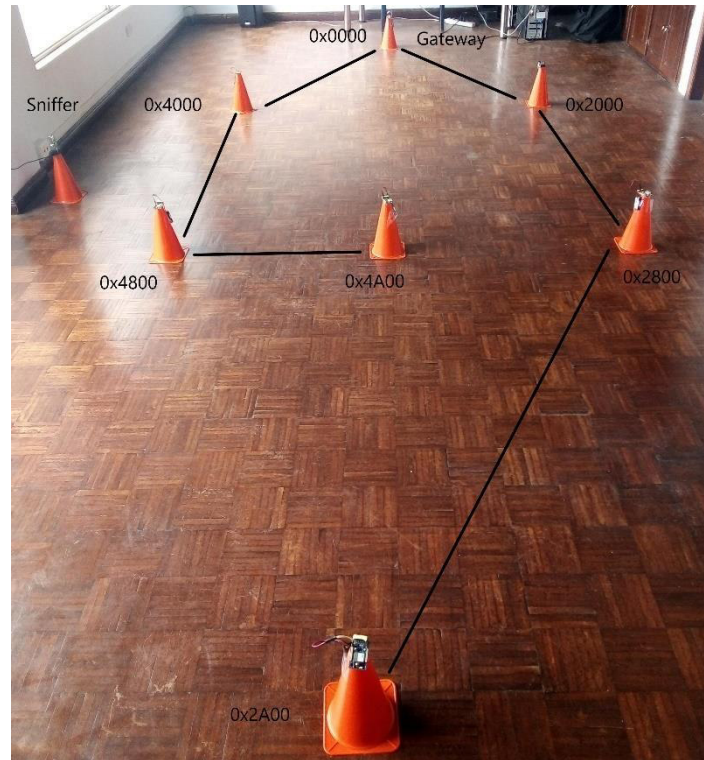


Figura 3.50. Topología final.

3.8. PRUEBA 8

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.51, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 7, en esta prueba se presenta un escenario con enlace ruidoso, el criterio para el proceso de asignación de direcciones será por número de saltos, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.52. Dado que es un escenario con enlace ruidoso se debe mencionar que el nodo configurado para simular dicho escenario es el que tiene la dirección aleatoria 0x40C6, como se explicó en Código 2.11 en el nodo mencionado la variable “enlaceRuidoso” tiene el valor de *true*.



Figura 3.51. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0	=0	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	118	OK	
1		16	DATA	0	0	0	1	0x00	0xCAFE	0x0265	0x0265		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6774243	=6774243	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	123	OK	
2		16	DATA	0	0	0	1	0x00	0xCAFE	0x40C6	0x40C6		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+7178486	=13952729	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	120	OK	
3		16	DATA	0	0	0	1	0x00	0xCAFE	0x5D7E	0x5D7E		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+6504472	=20457201	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	120	OK	
4		16	DATA	0	0	0	1	0x00	0xCAFE	0x5045	0x5045		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+7204322	=27661523	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	110	OK	
5		16	DATA	0	0	0	1	0x00	0xCAFE	0x335A	0x335A		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+5900883	=33562406	Type	Sec	Pnd	Ack.req	PAN_compr			MY_ID	107	OK	
6		16	DATA	0	0	0	1	0x00	0xCAFE	0x42AA	0x42AA		

Figura 3.52. Direcciones aleatorias asignadas a los nodos.

En la Figura 3.53 se observa que el mensaje “S” está acompañado de un asterisco, esto es debido a que se envió como valor hexadecimal en lugar de transformarlo a texto y después enviarlo, por lo que se hará uso de la Figura 3.54 para poder visualizar el valor de dicho asterisco que corresponde al número de saltos. Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “S01”, el nodo con el que se simula el enlace ruidoso no procesa esta primera trama y espera 1 segundo para enviar una trama

unicast con el mensaje “FALLO” al Gateway, como se observa en la trama 8 de la Figura 3.53 el nodo envía la trama después de alrededor de un segundo y medio, después el Gateway responde al nodo con una trama unicast y el mensaje “S01”, luego los dos nodos correspondientes al segundo nivel responden al Gateway con el mensaje “NEED_ID_S”, en el siguiente orden primero responde el nodo con la dirección 0x5D7E, luego el nodo con la dirección 0x40C6. De acuerdo con el orden de las tramas de la Figura 3.53, el Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que defina su dirección fija el nodo con la dirección 0x5D7E, después envía la información para que defina su dirección fija el nodo con la dirección 0x40C6, dado que el nodo con la dirección 0x5D7E calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “S02”, luego el nodo con la dirección 0x40C6 define su nueva dirección con el valor 0x4000 y envía una trama en broadcast con el mensaje “S02”.

P.nbr. RX 7	Time (us) +16223656 =49786062	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x0000	MAC payload S *	LQI 104	FCS OK
P.nbr. RX 8	Time (us) +1671834 =51457896	Length 16	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x40C6	MAC payload FAL LO	LQI 134	FCS OK
P.nbr. RX 9	Time (us) +2628 =51460524	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x40C6	Source Address 0x0000	MAC payload S *	LQI 107	FCS OK
P.nbr. RX 10	Time (us) +1402228 =52862752	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x5D7E	MAC payload NEED_ ID_S	LQI 153	FCS OK
P.nbr. RX 11	Time (us) +1763254 =54626006	Length 20	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x0000	Source Address 0x40C6	MAC payload NEED_ ID_S	LQI 131	FCS OK
P.nbr. RX 12	Time (us) +1585459 =56211465	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0x5D7E	Source Address 0x0000	MAC payload 1 *	LQI 85	FCS OK
P.nbr. RX 13	Time (us) +84641 =56296106	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x04	Dest. PAN 0xCAFE	Dest. Address 0x40C6	Source Address 0x0000	MAC payload 2 *	LQI 80	FCS OK
P.nbr. RX 14	Time (us) +1499 =56297605	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2000	MAC payload S *	LQI 153	FCS OK
P.nbr. RX 15	Time (us) +84258 =56381863	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4000	MAC payload S *	LQI 134	FCS OK

Figura 3.53. Asignación de direcciones del Gateway.

Como se puede observar en la Figura 3.54 se cumple el mismo proceso que se explicó en la PRUEBA 7 en la Figura 3.43, es decir, como en cada trama broadcast que envían los nodos asignadores se va aumentando el número de saltos.

P.nbr. RX 7	Time (us) +16223656 =49786062	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x01	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x0000	MAC payload 53 01	LQI 104	FCS OK
P.nbr. RX 9	Time (us) +2628 =51460524	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0x40C6	Source Address 0x0000	MAC payload 53 01	LQI 107	FCS OK
P.nbr. RX 14	Time (us) +1499 =56297605	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x02	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x2000	MAC payload 53 02	LQI 153	FCS OK
P.nbr. RX 15	Time (us) +84258 =56381863	Length 13	Frame control field Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	Sequence number 0x03	Dest. PAN 0xCAFE	Dest. Address 0xFFFF	Source Address 0x4000	MAC payload 53 02	LQI 134	FCS OK

Figura 3.54. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.55 que el nodo de tercer nivel con la dirección 0x335A responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x2000, después el nodo de tercer nivel con la dirección 0x5045 responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x4000, posteriormente el nodo asignador con la dirección 0x2000 envía información para que el nodo con dirección 0x335A calcule y defina su nueva dirección, también el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x5045 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x335A define su nueva dirección con el valor 0x2800 y envía una trama en broadcast con el mensaje “S03”, seguidamente el nodo con dirección 0x5045 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “S03”, después el nodo de cuarto nivel con la dirección 0x42AA responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x2800, posteriormente el nodo asignador con la dirección 0x2800 envía información para que el nodo con dirección 0x42AA calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x42AA define su nueva dirección con el valor 0x2A00 y envía una trama en broadcast con el mensaje “S04”. Aquí también se observa que cada vez que un nodo se convierte en nodo asignador cuando envía la trama broadcast el contador que acompaña a la letra “S” aumenta, como se explicó en Figura 3.43, esto se puede observar en la Figura 3.56.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3019519	20	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0x2000	0x335A	NEED_ID_S	107	OK
16	=59401382		DATA 0 0 0 1							
RX	+43860	20	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0x4000	0x5045	NEED_ID_S	126	OK
17	=59445242		DATA 0 0 0 1							
RX	+3118897	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x335A	0x2000	1	153	OK
18	=62564139		DATA 0 0 0 1							
RX	+53683	13	Type Sec Pnd Ack.req PAN_compr	0x04	0xCAFE	0x5045	0x4000	1	134	OK
19	=62617822		DATA 0 0 0 1							
RX	+32829	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x2800	S	107	OK
20	=62650651		DATA 0 0 0 1							
RX	+51433	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x4800	S	128	OK
21	=62702084		DATA 0 0 0 1							
RX	+3095746	20	Type Sec Pnd Ack.req PAN_compr	0x01	0xCAFE	0x2800	0x42AA	NEED_ID_S	69	OK
22	=65797830		DATA 0 0 0 1							
RX	+3192272	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x42AA	0x2800	1	110	OK
23	=68990102		DATA 0 0 0 1							
RX	+87063	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x2A00	S	72	OK
24	=69077165		DATA 0 0 0 1							

Figura 3.55. Asignación de direcciones de los nodos asignadores.

RX	+32829	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x2800	53	107	OK
20	=62650651		DATA 0 0 0 1							
RX	+51433	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x4800	53	128	OK
21	=62702084		DATA 0 0 0 1							
RX	+87063	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0xFFFF	0x2A00	53	72	OK
24	=69077165		DATA 0 0 0 1							

Figura 3.56. Mensaje con la letra "S", acompañada del valor del contador de saltos.

En este caso el escenario para probar el algoritmo es similar al escenario de nodo caído explicado en los anteriores criterios, es decir, se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.58, por lo tanto, una vez que el proceso de asignación de direcciones finalizó al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.57.

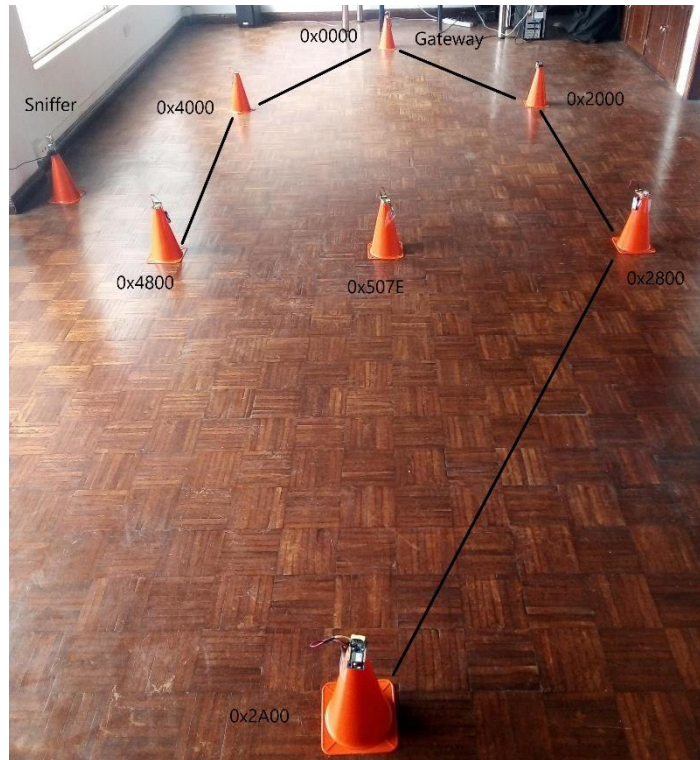


Figura 3.57. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+23320375		Type	Sec	Pnd	Ack. req	PAN_compr				MY_ID		
25	=92397540	16	DATA	0	0	0	1	0x00	0xCAFE	0x507E	0x507E	110	OK

Figura 3.58. Dirección aleatoria del nodo que se encendió.

Cuando el nodo se enciende, define su dirección aleatoria con el valor de 0x507E, aunque el proceso es parecido al del escenario de nodo caído en este caso el nodo envía una trama broadcast con el mensaje “SALTO”, los nodos asignadores que se encuentren dentro de la zona de cobertura del nodo le responden con una trama unicast, en este escenario el nodo responderá la oferta del nodo asignador por el que menos número de saltos para llegar al Gateway tenga que pasar, en este caso el nodo asignador con la dirección 0x4800 le respondió con el mensaje “S03” al nodo con la dirección 0x507E, luego el nodo con la dirección 0x507E le responde al nodo asignador con el mensaje “NEED_ID_S”. Hecho esto el nodo asignador le envía al nodo la información para que calcule y defina su nueva dirección, luego de esto el nodo define su nueva dirección con el valor de 0x4A00 se convierte en nodo asignador y envía una trama broadcast con el mensaje “S04”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4068581		Type	Sec	Pnd	Ack.req	PAN_compr				SAL		
26	=96466121	16	DATA	0	0	0	1	0x01	0xC9FE	0xFFFF	0x507E	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3300		Type	Sec	Pnd	Ack.req	PAN_compr				S		
27	=96469421	13	DATA	0	0	0	1	0x03	0xC9FE	0x507E	0x4800	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3143709		Type	Sec	Pnd	Ack.req	PAN_compr				NEED_		
28	=99613130	20	DATA	0	0	0	1	0x02	0xC9FE	0x4800	0x507E	83	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3148375		Type	Sec	Pnd	Ack.req	PAN_compr				1		
29	=102761505	13	DATA	0	0	0	1	0x04	0xC9FE	0x507E	0x4800	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+86484		Type	Sec	Pnd	Ack.req	PAN_compr				S		
30	=102847989	13	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	0x4A00	80	OK

Figura 3.59. Asignación de dirección al nodo para probar escenario.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3300		Type	Sec	Pnd	Ack.req	PAN_compr				53		
27	=96469421	13	DATA	0	0	0	1	0x03	0xC9FE	0x507E	0x4800	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+86484		Type	Sec	Pnd	Ack.req	PAN_compr				53		
30	=102847989	13	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	0x4A00	80	OK

Figura 3.60. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.61, además una vez finalizado este proceso de acuerdo con la Figura 3.59 el tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 102.85 segundos.

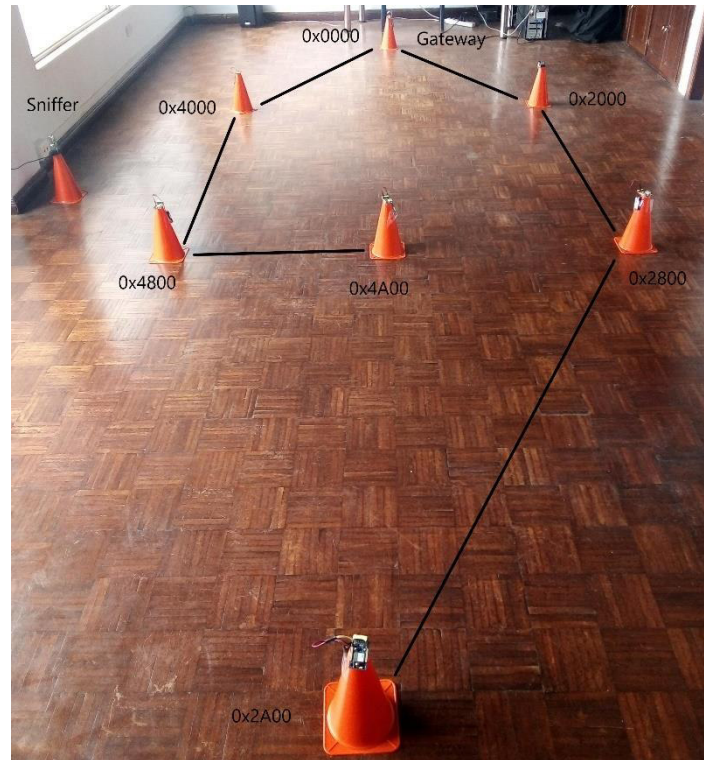


Figura 3.61. Topología final.

3.9. PRUEBA 9

Para la realización de esta prueba los nodos se ubicaron en las posiciones mostradas en la Figura 3.62, los nodos se encuentran a las mismas distancias que las mencionadas en la PRUEBA 7, en esta prueba se presenta un escenario con nodo caído, el criterio para el proceso de asignación de direcciones será por número de saltos, las direcciones aleatorias asignadas a los nodos se definen de acuerdo con los mostrado en la Figura 3.63. Dado que es un escenario con nodo caído se debe mencionar que el nodo configurado para simular dicho escenario es el que no tiene dirección aleatoria y se muestra como “APAGADO” en la Figura 3.62 ya que al inicio se encuentra apagado y una vez finalizado el proceso de asignación de direcciones se encendió manualmente para probar el escenario con nodo caído, como se explicó en Código 2.11 en el nodo mencionado la variable “nodoCaido” tiene el valor de *true*. Cabe aclarar que en la Figura 3.62 el nodo como el mensaje “Nodo para probar algoritmo (apagado)” es muy diferente al nodo con mensaje “APAGADO” ya que el primero que se mencionó se utilizó para probar el escenario de número de saltos, mientras que el otro nodo mencionado se usó para probar el escenario de nodo caído.



Figura 3.62. Ubicación de los nodos antes de la asignación de direcciones fijas.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+0		Type	Sec	Pnd	Ack.req	PAN_compr				MY_		
1	=0	16	DATA	0	0	0	1	0x00	0xCAFE	0x76BD	ID	120	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3247566		Type	Sec	Pnd	Ack.req	PAN_compr				MY_		
2	=3247566	16	DATA	0	0	0	1	0x00	0xCAFE	0x2C61	ID	134	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4800699		Type	Sec	Pnd	Ack.req	PAN_compr				MY_		
3	=8048265	16	DATA	0	0	0	1	0x00	0xCAFE	0x7C56	ID	123	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+4830929		Type	Sec	Pnd	Ack.req	PAN_compr				MY_		
4	=12879194	16	DATA	0	0	0	1	0x00	0xCAFE	0x0DD7	ID	126	OK
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3799149		Type	Sec	Pnd	Ack.req	PAN_compr				MY_		
5	=16678343	16	DATA	0	0	0	1	0x00	0xCAFE	0x2BC7	ID	102	OK

Figura 3.63. Direcciones aleatorias asignadas a los nodos.

En la Figura 3.64 se observa que el mensaje “S” está acompañado de un asterisco, esto es debido a que se envió como valor hexadecimal en lugar de transformarlo a texto y después enviarlo, por lo que se hará uso de la Figura 3.65 para poder visualizar el valor de dicho asterisco que corresponde al número de saltos. Una vez iniciado el proceso de asignación de direcciones el Gateway define su dirección con el valor de 0x0000 y envía una trama broadcast (0xFFFF) con el mensaje “S01”, luego los dos nodos correspondientes al segundo nivel responden al Gateway con el mensaje “NEED_ID_S”, en el siguiente orden primero responde el nodo con la dirección 0x7C56, luego el nodo con la dirección 0x2C61. De acuerdo con el orden de las tramas de la Figura 3.64, el

Gateway respondió según el orden en que los nodos le respondieron, primero envía la información para que defina su dirección fija el nodo con la dirección 0x7C56, después envía la información para que defina su dirección fija el nodo con la dirección 0x2C61, dado que el nodo con la dirección 0x7C56 calcula y define su nueva dirección con el valor de 0x2000 se convierte en nodo asignador y envía una trama en broadcast con el mensaje “S02”, luego el nodo con la dirección 0x2C61 define su nueva dirección con el valor 0x4000 y envía una trama en broadcast con el mensaje “S02”.

P.nbr.	Time [us]	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 6	+10442274 =27120617	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0000	S *	110	OK
RX 7	+3077665 =30198282	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x7C56	NEED_ ID_S	153	OK
RX 8	+11176 =30209458	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x0000	0x2C61	NEED_ ID_S	139	OK
RX 9	+3340309 =33549767	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0x7C56	0x0000	1 *	53	OK
RX 10	+84062 =33633829	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x2C61	0x0000	2 *	53	OK
RX 11	+828 =33634657	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x2000	S *	155	OK
RX 12	+84426 =33719083	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4000	S *	136	OK

Figura 3.64. Asignación de direcciones del Gateway.

Como se puede observar en la Figura 3.65 se cumple el mismo proceso que se explicó en la PRUEBA 7 en la Figura 3.43, es decir, como en cada trama broadcast que envían los nodos asignadores se va aumentando el número de saltos.

P.nbr.	Time [us]	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 6	+10442274 =27120617	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0xFFFF	0x0000	53 01	110	OK
RX 11	+828 =33634657	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x2000	53 02	155	OK
RX 12	+84426 =33719083	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4000	53 02	136	OK

Figura 3.65. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez que los nodos pasaron a ser nodos asignadores y enviaron las tramas broadcast para asignar direcciones a los nodos que se encuentran dentro de su zona de cobertura, se observa de acuerdo con la Figura 3.66 que el nodo de tercer nivel con la dirección 0x0DD7 responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección

0x4000, posteriormente el nodo asignador con la dirección 0x4000 envía información para que el nodo con dirección 0x0DD7 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x0DD7 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “S03”, seguidamente el nodo con dirección 0x0DD7 define su nueva dirección con el valor 0x4800 y envía una trama en broadcast con el mensaje “S03”, después el nodo de cuarto nivel con la dirección 0x2BC7 responde con el mensaje “NEED_ID_S” al nodo asignador con la dirección 0x4800, posteriormente el nodo asignador con la dirección 0x4800 envía información para que el nodo con dirección 0x2BC7 calcule y defina su nueva dirección, hecho esto el nodo con dirección 0x2BC7 define su nueva dirección con el valor 0x4A00 y envía una trama en broadcast con el mensaje “S04”. Aquí también se observa que cada vez que un nodo se convierte en nodo asignador cuando envía la trama broadcast el contador que acompaña a la letra “S” aumenta, como se explicó en Figura 3.43, esto se puede observar en la Figura 3.66.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 13	+3063081 =36782164	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4000	0x0DD7	NEED_ID_S	128	OK
RX 14	+3174512 =39956676	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x0DD7	0x4000	1 *	139	OK
RX 15	+84170 =40040846	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4800	S *	123	OK
RX 16	+3144461 =43185307	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4800	0x2BC7	NEED_ID_S	67	OK
RX 17	+3150784 =46336091	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x2BC7	0x4800	1 -	123	OK
RX 18	+88266 =46424357	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4A00	S *	67	OK

Figura 3.66. Asignación de direcciones de los nodos asignadores.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 15	+84170 =40040846	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4800	53 03	123	OK
RX 18	+88266 =46424357	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4A00	53 04	67	OK

Figura 3.67. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

En este caso el escenario para probar el algoritmo es similar al escenario de nodo caído explicado en los anteriores criterios, es decir, se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.69, por lo tanto, una vez que el proceso de

asignación de direcciones finalizó al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.68.

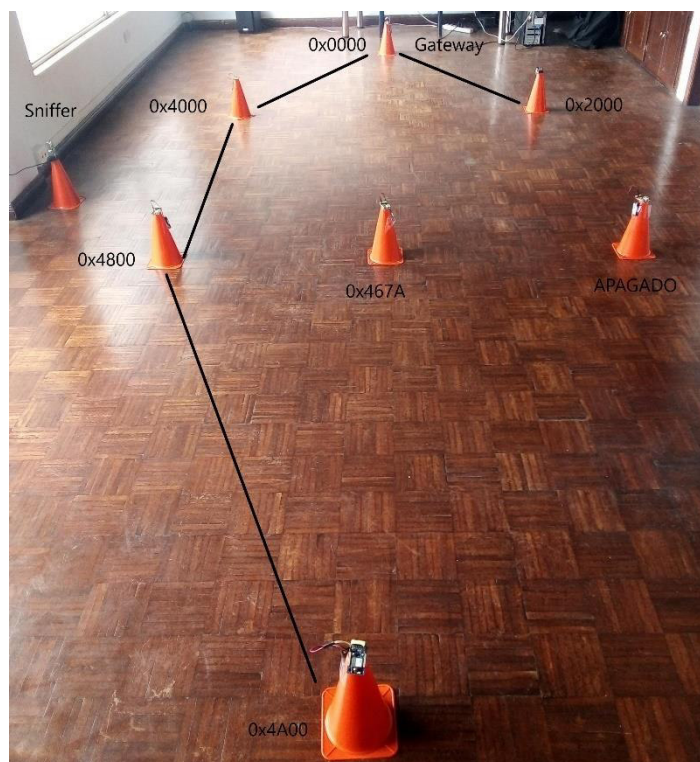


Figura 3.68. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+23972634	16	Type	Sec	Prd	Ack.req	PAN_compr				MY		
19	=70396991	16	DATA	0	0	0	1	0x00	0xCAFE	0x467A	ID	131	OK

Figura 3.69. Dirección aleatoria del nodo que se encendió.

Cuando el nodo se enciende, define su dirección aleatoria con el valor de 0x467A, aunque el proceso es parecido al del escenario de nodo caído en este caso el nodo envía una trama broadcast con el mensaje “SALTO”, los nodos asignadores que se encuentren dentro de la zona de cobertura del nodo le responden con una trama unicast, en este escenario el nodo responderá la oferta del nodo asignador por el que menos número de saltos para llegar al Gateway tenga que pasar, en este caso el nodo asignador con la dirección 0x2000 le respondió con el mensaje “S02” al nodo con la dirección 0x467A, luego el nodo con la dirección 0x467A le responde al nodo asignador con el mensaje “NEED_ID_S”. Hecho esto el nodo asignador le envía al nodo la información para que calcule y defina su nueva dirección, luego de esto el nodo define su nueva dirección con el valor de 0x2800 se convierte en nodo asignador y envía una trama broadcast con el mensaje “S03”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+15171510	16	Type	Sec	Pnd	Ack.req	PAN_compr	0x01	0xCAFE	0xFFFF	0x467A	136	OK
20	=85568501		DATA	0	0	0	1						
										SAL			
										TO			

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+1689	13	Type	Sec	Pnd	Ack.req	PAN_compr	0x03	0xCAFE	0x467A	0x2000	155	OK
21	=85570190		DATA	0	0	0	1						
										S			
										*			

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3149416	20	Type	Sec	Pnd	Ack.req	PAN_compr	0x02	0xCAFE	0x2000	0x467A	136	OK
22	=88719606		DATA	0	0	0	1						
										NEED_			
										ID_S			

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3162818	13	Type	Sec	Pnd	Ack.req	PAN_compr	0x04	0xCAFE	0x467A	0x2000	155	OK
23	=91882424		DATA	0	0	0	1						
										1			
										*			

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84592	13	Type	Sec	Pnd	Ack.req	PAN_compr	0x03	0xCAFE	0xFFFF	0x2800	136	OK
24	=91967016		DATA	0	0	0	1						
										S			
										*			

Figura 3.70. Asignación de dirección al nodo para probar escenario.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+1689	13	Type	Sec	Pnd	Ack.req	PAN_compr	0x03	0xCAFE	0x467A	0x2000	155	OK
21	=85570190		DATA	0	0	0	1						
										53			
										02			

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84592	13	Type	Sec	Pnd	Ack.req	PAN_compr	0x03	0xCAFE	0xFFFF	0x2800	136	OK
24	=91967016		DATA	0	0	0	1						
										53			
										03			

Figura 3.71. Mensaje con la letra "S", acompañada del valor del contador de saltos.

Para probar el escenario de nodo caído se debe encender el nodo para realizar el procedimiento de ingreso a la red, una vez encendido el nodo se definirá una dirección aleatoria de acuerdo con la Figura 3.73, por lo tanto, una vez que el proceso de asignación de direcciones hasta que el nodo para probar el algoritmo finalizó, al encender el nuevo nodo se obtendrá la topología mostrada en la Figura 3.72.



Figura 3.72. Topología con la dirección del nodo que se encendió.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX			Type	Sec	Pnd	Ack.req	PAN_compr						
25	+13443105 =105410121	16	DATA	0	0	0	1	0x00	0xC9FE	0x1D7B	MY_ID	107	OK

Figura 3.73. Dirección aleatoria del nodo que se encendió.

Una vez que el nodo se enciende y define su dirección aleatoria con el valor de 0x1D7B, envía una trama broadcast con el mensaje “CAIDO”, como se observa en la Figura 3.74 le responden los nodos asignadores con las direcciones 0x4A00 y 0x2800 con los mensajes “S04” y “S03” respectivamente, de acuerdo con el criterio de número de saltos el nodo responderá al nodo asignador con menor número de saltos para llegar al Gateway, es decir, le responde al nodo asignador con dirección 0x2800, posteriormente dicho nodo asignador le envía la información para calcular y definir su nueva dirección, hecho esto el nodo define su nueva dirección con el valor de 0x2A00 y envía una trama broadcast con el mensaje “S04”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX			Type	Sec	Pnd	Ack.req	PAN_compr						
26	+4912094 =110322215	16	DATA	0	0	0	1	0x01	0xC9FE	0xFFFF	CAI DO	96	OK
27	+1709 =110323924	13	DATA	0	0	0	1	0x03	0xC9FE	0x1D7B	S*	51	OK
28	+1262 =110325186	13	DATA	0	0	0	1	0x04	0xC9FE	0x1D7B	S*	136	OK
29	+3184042 =113509228	20	DATA	0	0	0	1	0x02	0xC9FE	0x2800	NEED_ID S	104	OK
30	+3152367 =116661595	13	DATA	0	0	0	1	0x05	0xC9FE	0x1D7B	1-	139	OK
31	+85597 =116747192	13	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	S*	107	OK

Figura 3.74. Asignación de dirección al nodo caído.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX			Type	Sec	Pnd	Ack.req	PAN_compr						
27	+1709 =110323924	13	DATA	0	0	0	1	0x03	0xC9FE	0x1D7B	53 04	51	OK
28	+1262 =110325186	13	DATA	0	0	0	1	0x04	0xC9FE	0x1D7B	53 03	136	OK
31	+85597 =116747192	13	DATA	0	0	0	1	0x03	0xC9FE	0xFFFF	53 04	107	OK

Figura 3.75. Mensaje con la letra “S”, acompañada del valor del contador de saltos.

Una vez finalizado el proceso de asignación de direcciones, todos los nodos se encuentran correctamente identificados y se obtiene la topología que se observa en la Figura 3.76, además una vez finalizado este proceso de acuerdo con la Figura 3.74 el

tiempo que le tomó a la red que todos sus nodos obtengan sus respectivas tablas de enrutamiento es de 116.75 segundos.

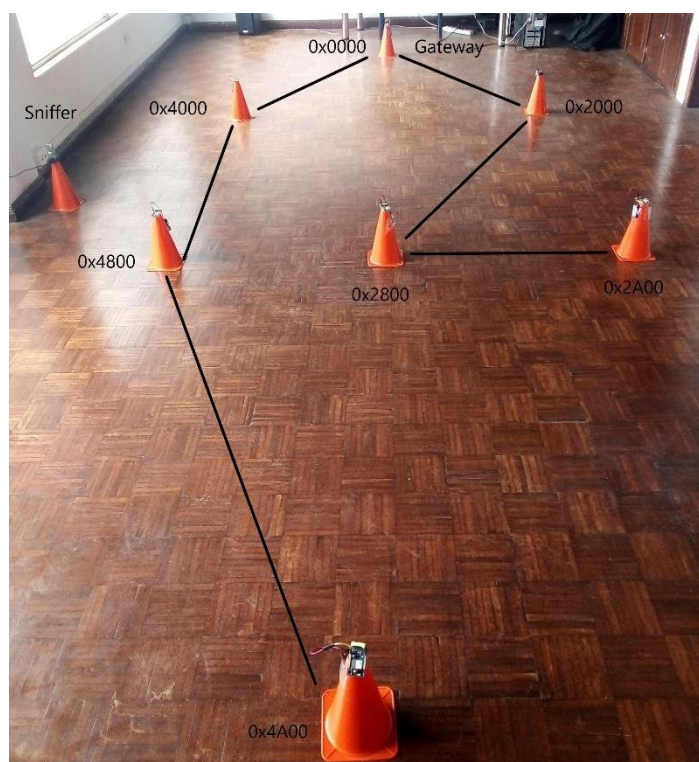


Figura 3.76. Topología final.

3.10. PRUEBA 10

Para probar que la tabla de enrutamiento en cada nodo se haya formado correctamente se realizaron varias pruebas en las distintas topologías y usando los diferentes criterios para el proceso de asignación de direcciones. Se obtuvieron los mejores resultados en cuanto a tiempos con el criterio de nivel de potencia de la trama recibida.

Para el proceso de comunicación de extremo a extremo cabe mencionar que esto implica, tanto la comunicación entre un nodo y el Gateway o entre dos nodos de la red. Dado que el Gateway básicamente es otro nodo de la red se utilizó el mismo código para los dos casos que se mencionaron. Por lo que a continuación se presenta un ejemplo de prueba realizada de comunicación entre un nodo y el Gateway. Como se explicó en las anteriores pruebas en la Figura 3.77 se observan las direcciones aleatorias que obtendrán los nodos al ser encendidos.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX			Type	Sec	Pnd	Ack.req	PAN_compr			MY_			
			DATA	0	0	0	1	0x00	0xC9FE	0x6F9F	ID	115	OK
1	+0	=0	16										
2	+6271466	=6271466	16					0x00	0xC9FE	0x7C56	ID	120	OK
3	+9492987	=15764453	16					0x00	0xC9FE	0x5AAC	ID	93	OK
4	+8946409	=24710862	16					0x00	0xC9FE	0x43FF	ID	158	OK
5	+6555562	=31266424	16					0x00	0xC9FE	0x6D35	ID	142	OK
6	+6541848	=37808272	16					0x00	0xC9FE	0x3E36	ID	139	OK
7	+5530623	=43338895	16					0x00	0xC9FE	0x0F9A	ID	110	OK

Figura 3.77. Direcciones aleatorias asignadas a los nodos.

En la Figura 3.78 se observa como el Gateway envía una trama broadcast con el mensaje “P” a la cual responden los dos nodos ubicados en el segundo nivel, luego el Gateway les proporciona información para que calculen y definan su nueva dirección, hecho esto se convierten en nodos asignadores al enviar una trama broadcast con el mensaje “P”.

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX			Type	Sec	Pnd	Ack.req	PAN_compr			P			
			DATA	0	0	0	1	0x01	0xC9FE	0xFFFF	0x0000	91	OK
8	+12392037	=55730932	12										
9	+3076597	=58807529	20					0x01	0xC9FE	0x0000	NEED_ID_P	161	OK
10	+8268	=58815797	20					0x01	0xC9FE	0x0000	NEED_ID_P	134	OK
11	+3339986	=62155793	13					0x02	0xC9FE	0x5AAC	1	93	OK
12	+82667	=62238450	13					0x03	0xC9FE	0x7C56	2	93	OK
13	+2113	=62240563	12					0x02	0xC9FE	0xFFFF	P	161	OK
14	+82737	=62323300	12					0x02	0xC9FE	0xFFFF	P	134	OK

Figura 3.78. Asignación de direcciones del Gateway.

En la Figura 3.79 se observa como los nodos de tercer nivel responden a las tramas broadcast enviadas por los nodos asignadores, se puede ver que tres nodos responden al nodo asignador con la dirección 0x2000 mientras que un nodo responde al nodo

asignador con la dirección 0x4000. Luego de esto cada nodo asignador envía información para que calculen y definan su nueva dirección a los nodos que les corresponda, hecho esto los nodos definen su nueva dirección y se convierten en nodos asignadores para luego enviar una trama broadcast con el mensaje “P”.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX 15	+2980407 =65303707	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x2000	0x6D35	NEED_ID_P	134	OK
RX 16	+1313 =65305020	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x2000	0x3E36	NEED_ID_P	131	OK
RX 17	+34886 =65339906	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x2000	0x0F9A	NEED_ID_P	99	OK
RX 18	+123370 =65463276	20	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x01	0xCAFE	0x4000	0x43FF	NEED_ID_P	139	OK
RX 19	+3005771 =68469047	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x6D35	0x2000	1 *	161	OK
RX 20	+77521 =68546568	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x04	0xCAFE	0x3E36	0x2000	2 *	161	OK
RX 21	+8831 =68555399	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x2800	P	134	OK
RX 22	+75617 =68631016	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x05	0xCAFE	0x0F9A	0x2000	3 *	161	OK
RX 23	+1311 =68632327	13	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x03	0xCAFE	0x43FF	0x4000	1 *	134	OK
RX 24	+4404 =68636731	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x3000	P	128	OK
RX 25	+81752 =68718483	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x3800	P	96	OK
RX 26	+1441 =68719924	12	Type Sec Pnd Ack.req PAN_compr DATA 0 0 0 1	0x02	0xCAFE	0xFFFF	0x4800	P	139	OK

Figura 3.79. Asignación de direcciones de los nodos asignadores.

En la Figura 3.80 se observa la topología final con los transceptores correctamente identificados una vez que finalizó el proceso de asignación de direcciones.

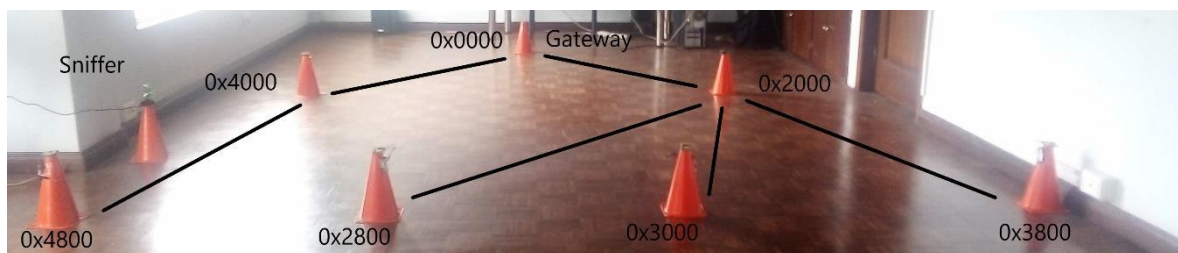


Figura 3.80. Topología final.

En la Figura 3.81 se observa que, una vez finalizado el proceso de asignación de direcciones, se puede iniciar el proceso de comunicación. En la Figura 3.80 en la esquina inferior izquierda se tiene el nodo asignador con la dirección 0x4800, previamente este nodo se configuró para dar inicio al proceso de comunicación, definiendo la dirección de destino con el valor de 0x0000 (Gateway), es decir, al presionar el botón SW2 del transceptor revisa si la dirección de destino es la suya, como no lo es envía a su nodo asignador (0x4800) una trama unicast con el mensaje “ENVIO0000”, el mensaje está compuesto por el texto “ENVIO” y la dirección de destino en texto en este caso “0000”, luego el nodo asignador con la dirección 0x4800 revisa si la dirección destino es la suya, como no lo es envía una trama unicast con el mismo mensaje que recibió del transceptor con la dirección 0x4000 a su nodo asignador que en este caso es el Gateway (0x0000).

P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3743185	=0	Type	Sec	Pnd	Ack.req	PAN_compr	0x03	0xC9FE	0x4000	ENVIO	147	OK
27		20	DATA	0	0	0	1			0x4800	0***		
P.nbr.	Time (us)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+86013	=86013	Type	Sec	Pnd	Ack.req	PAN_compr	0x04	0xC9FE	0x0000	ENVIO	131	OK
28		20	DATA	0	0	0	1			0x4000	0***		

Figura 3.81. Proceso de comunicación del nodo al Gateway.

En la Figura 3.82 se observa el camino que sigue el mensaje de comunicación desde el transceptor que inicia el proceso hasta que llega a su destino, a través de las flechas verdes, es decir, iniciando en el transceptor con dirección 0x4800, pasando por el transceptor con dirección 0x4000 y finalizando al llegar al Gateway con dirección 0x0000.

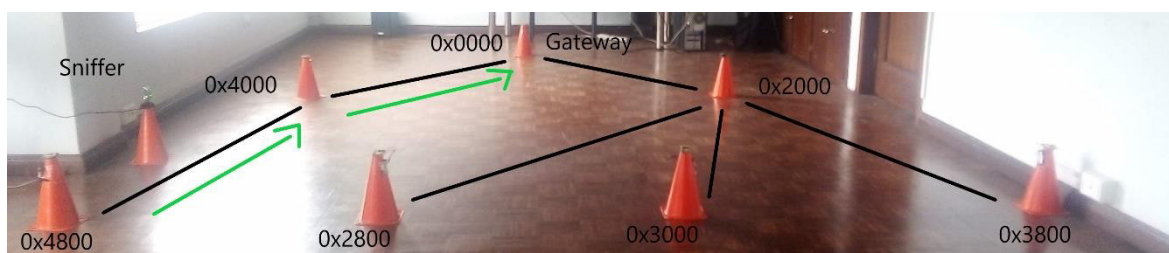


Figura 3.82. Topología con la ruta de comunicación del nodo al Gateway.

3.11. ANÁLISIS DE RESULTADOS OBTENIDOS

De acuerdo al número de transceptores que se usaron y según lo observado en cada una de las pruebas el máximo número de nodos al que un nodo asignador le proporcionó información para que calculen y definan su nueva dirección es tres por lo que en todos los casos el nodo asignador usó dos bits, por lo tanto, a partir del Gateway el número de bits disponibles que se envían va disminuyendo en un valor de dos, es decir, el Gateway al iniciar el proceso de asignación de direcciones tiene un valor de bits disponibles de 0xF

en hexadecimal que corresponde a 15 en decimal, para asignar direcciones utiliza dos bits por los nodos asignadores de segundo nivel el valor de bits disponibles es de 0D en valor hexadecimal que corresponde a 13 decimal como se observa en la Figura 3.83.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3335925	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0x44EB	0x0000	1	72	OK
10	=39950616		DATA 0 0 0 1					*		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84521	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x7228	0x0000	2	72	OK
11	=40035137		DATA 0 0 0 1					*		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3335925	13	Type Sec Pnd Ack.req PAN_compr	0x02	0xCAFE	0x44EB	0x0000	31	72	OK
10	=39950616		DATA 0 0 0 1					0D		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+84521	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x7228	0x0000	32	72	OK
11	=40035137		DATA 0 0 0 1					0D		

Figura 3.83. Valores enviados por los nodos asignadores a los nodos.

Cuando los transeptores de segundo nivel asignan direcciones usan dos bits por lo que para los nodos asignadores de tercer nivel el valor de bits disponibles es de 0B hexadecimal que corresponde a 11 decimal como se observa en la Figura 3.84.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3111677	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x3F34	0x2000	1	155	OK
16	=46297504		DATA 0 0 0 1					*		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+55846	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x2CBE	0x4000	1	131	OK
17	=46353350		DATA 0 0 0 1					*		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3111677	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x3F34	0x2000	31	155	OK
16	=46297504		DATA 0 0 0 1					0B		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+55846	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x2CBE	0x4000	31	131	OK
17	=46353350		DATA 0 0 0 1					0B		

Figura 3.84. Valores enviados por los nodos asignadores a los nodos.

Finalmente, los transeptores de tercer nivel usan dos bits para asignar direcciones por lo que el valor de bits disponibles para los nodos asignadores de cuarto nivel es de 09 hexadecimal que corresponde a 9 decimal como se observa en la Figura 3.85.

P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3185480	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x709D	0x2800	1	107	OK
21	=52715030		DATA 0 0 0 1					-		
P.nbr.	Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	MAC payload	LQI	FCS
RX	+3185480	13	Type Sec Pnd Ack.req PAN_compr	0x03	0xCAFE	0x709D	0x2800	31	107	OK
21	=52715030		DATA 0 0 0 1					09		

Figura 3.85. Valores enviados por los nodos asignadores a los nodos.

Cabe aclarar que para el análisis de los resultados obtenidos se tomará en cuenta al escenario con un ambiente ideal, es decir, sin fallos como el mejor tiempo que se logra obtener.

Una vez finalizadas las pruebas que se realizaron se puede analizar los resultados obtenidos.

En general el algoritmo cumple con el objetivo de la obtención de la tabla de enrutamiento mediante la asignación de direcciones a nivel de capa enlace de forma jerárquica con cada uno de los tres criterios que se han mencionado, logrando así obtener una topología tipo árbol, aunque como se observó en cada una de las pruebas según el criterio que se use la configuración de la red cambia, además permite la comunicación entre un nodo y el Gateway.

Tabla 3.1. Tiempos obtenidos en el proceso de asignación de direcciones.

	Escenario sin fallos	Escenario con enlaces ruidosos	Escenario con nodos caídos
Criterio de orden de llegada	57.16 seg	61.54 seg	96.38 seg
Criterio de nivel de potencia	50.34 seg	62.38 seg	118.26 seg
Criterio de número de saltos	86.13 seg	102.85 seg	116.75 seg

Para el proceso de asignación de direcciones para cada uno de los criterios utilizados se simuló un escenario sin fallos, un escenario con enlaces ruidosos y un escenario con nodos caídos. Para el escenario sin fallos el mejor tiempo se obtuvo con el criterio de nivel de potencia de la trama recibida con un valor de 50.34 segundos. Para el escenario con enlaces ruidosos el mejor tiempo se obtuvo con el criterio de orden de llegada de la oferta de asignación con un valor de 61.54 segundos, aunque muy similar al tiempo obtenido con el criterio de nivel de potencia de la trama recibida con un valor de 62.83 segundos. Para el escenario de nodos caídos el mejor tiempo se obtuvo con el criterio de orden de llegada de la oferta de asignación con un valor de 96.38 segundos.

En el caso de probar la comunicación entre un nodo y el Gateway una vez obtenida la tabla de enrutamiento se realizaron varias pruebas con los diferentes criterios y se obtuvo el mejor resultado con el criterio de nivel de potencia de la trama recibida en un escenario

sin fallos consiguiendo un tiempo de 68.72 segundos en la obtención de la tabla de enrutamiento y un tiempo de 86.01 milisegundos en el proceso de comunicación.

Según los tiempos obtenidos en las pruebas realizadas se puede decir que la obtención de la tabla de enrutamiento tanto por el criterio de orden de llegada de la oferta de asignación como por el criterio de nivel de potencia de la trama recibida son óptimos ya que se obtuvo tiempos similares. Además, se observó que el criterio de número de saltos no resulta tan bueno ya que para este caso se creó un escenario similar al escenario de nodos caídos que permita ver el correcto funcionamiento del algoritmo por lo que se obtienen tiempos más altos en relación con los otros dos criterios.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Los tiempos obtenidos en cada una de las pruebas y el análisis de estos permiten observar que los criterios óptimos, es decir, en los que se consiguió los mejores tiempos de obtención de la tabla de enrutamiento son el criterio de orden de llegada de la oferta de asignación y el criterio de nivel de potencia de la trama recibida.

El ambiente en el que se realizan las pruebas influye de gran manera ya que los transceptores son bastante sensibles a la interferencia provocada por otras señales que también trabajan en la banda de frecuencia de los 2.4 GHz, además los obstáculos con los que las señales se puedan encontrar son otro aspecto para tener en cuenta. En sí cualquier fuente que produzca ruido producirá errores ya sea en mayor o menor medida al momento de probar el funcionamiento del algoritmo.

En el proceso de asignación de direcciones para la obtención de la tabla de enrutamiento se simuló tanto escenario sin fallos, escenario con enlaces ruidosos y escenario con nodos caídos, con el fin de que sea un proceso robusto, es decir, se trató de minimizar al máximos los errores que se puedan producir, pero para el proceso de comunicación simplemente se envía la información por lo que en varias pruebas de las que se realizaron se produjeron errores al momento de enviar información desde un nodo al Gateway o desde un nodo a otro nodo.

Debido a que los nodos asignadores proporcionan información a los transceptores (nodos) que se encuentran dentro de su zona de cobertura fue necesario definir un rango de distancia para que solo ciertos transceptores recepten y procesen dichas tramas con el fin de probar el funcionamiento del algoritmo. Por lo que se trabajó con ciertos parámetros como: potencia de transmisión de 3 dBm, sensibilidad de -54 dBm, rango de cobertura de 2.5 metros, para la comunicación se utilizó el canal 20 y los transceptores se colocaron a una altura de 23 cm ya que se utilizaron conos para probar el prototipo de red.

Se mencionó que las pruebas se realizarían con al menos 7 transceptores por lo que se utilizaron 6 transceptores ATZB-256RFR2-XPRO y 1 transceptor RCB256RFR2, una vez que se analizaron los manuales se observó que básicamente los dos tipos tienen la misma sintaxis al momento de codificar por lo que no hubo inconveniente en usar el mismo código, la única diferencia notable es el dispositivo para grabar el código en los

transceptores. Los ATZB-256RFR2-XPRO usan un dispositivo ATMEL-ICE mientras que los RCB256RFR2 usan un dispositivo AVR-DRAGON.

Las pruebas realizadas permiten observar que la topología final que se obtiene una vez que los transceptores están correctamente identificados es de tipo árbol, pero dependiendo del criterio que se utilice para el proceso de asignación de direcciones para la obtención de la tabla de enrutamiento la configuración de dicha topología cambia.

Los transceptores ATZB-256RFR2-XPRO constan de tres LEDs (rojo, verde y amarillo) y dos botones (SW1 y SW2), esto resultó muy útil al momento de realizar el código en Atmel Studio 7.0 ya que en caso de que no se esté ejecutando alguna parte del código se puede encender o apagar uno o varios LEDs para identificar donde se está dando la falla y proceder a corregirla, además el botón SW1 permite reiniciar cualquier proceso y es una tarea que ya está configurada por defecto en el transceptor, pero al botón SW2 se le puede configurar una diferente tarea a medida que se presiona el botón SW2 más de una vez, esto permite controlar el inicio del proceso de asignación de direcciones y comunicación.

La herramienta que se utiliza como Sniffer o capturador de tramas (Texas Instruments SmartRF Packet Sniffer) también ocupa un cierto tiempo hasta procesar las tramas y mostrarlas en pantalla, por lo que en ciertas pruebas realizadas no se mostraban una o varias tramas, pero esto no quiere decir que haya fallado el algoritmo sino simplemente que el Sniffer no las capturó, por lo que si se vuelve a realizar la prueba se podrán visualizar dichas tramas.

El proceso de asignación de direcciones para identificar correctamente a los transceptores se realizó a nivel de capa enlace con el fin de minimizar el uso de recursos y procesamiento, ya que los transceptores que se utilizó trabajan con pilas por lo que minimizar el consumo de energía es de suma importancia para aumentar tanto la vida útil de los transceptores como el tiempo funcionamiento de la red antes de que se tenga que reemplazar la fuente de energía.

Mediante la realización de las pruebas y el análisis de los resultados se comprobó el funcionamiento del algoritmo propuesto, ya que el proceso de asignación de direcciones para la obtención de la tabla de enrutamiento se realiza de manera correcta, es decir, de forma jerárquica empezando por el Gateway que define su dirección en 0x0000 y proporciona información a los nodos que se encuentran dentro de su zona de cobertura para que calculen y definan su nueva dirección, este proceso se repetirá sucesivamente

hasta que todos los nodos se encuentren correctamente identificados y así obtener una topología tipo árbol.

Debido a que el consumo de energía es un aspecto importante para tomar en cuenta en las redes inalámbricas de sensores, el uso del estándar IEEE 802.15.4 es de gran ayuda ya que permite utilizar el direccionamiento corto, es decir, que solamente utiliza 16 bits para el proceso de asignación de direcciones en lugar de los 64 bits que se podría utilizar, logrando así minimizar el procesamiento que realizan los transceptores.

4.2. RECOMENDACIONES

Dado que se están manejando tres criterios de obtención de la tabla de enrutamiento y en el código esto se controla mediante variables booleanas es necesario revisar que las variables tengan el valor correcto (*true* o *false*) antes de grabar el código en los transceptores.

El Sniffer o capturador de tramas (Texas Instruments SmartRF Packet Sniffer) de ser posible se debe colocar en la parte central de la red con el fin de que sea capaz de procesar todas las tramas que se están transmitiendo y recibiendo entre los transceptores, con el fin de minimizar errores de procesamiento y visualización de tramas.

Las baterías de los transceptores deben estar cargadas con alrededor de 2.7 a 3 voltios ya que al tener menos voltaje afecta al correcto funcionamiento del algoritmo, porque puede provocar fallos al enviar o recibir tramas.

Con el fin de minimizar errores al momento de la transmisión y recepción de tramas se deben colocar los transceptores en una posición en la que exista línea de vista entre todos los que componen la red WSN.

Es necesario que la etapa de asignación de direcciones para la obtención de la tabla de enrutamiento finaliza correctamente, es decir, que todos los transceptores estén correctamente identificados para que se pueda iniciar la etapa de comunicación.

Al momento de crear el proyecto en Atmel Studio 7.0 se debe elegir correctamente el microcontrolador con el que trabajan los transceptores que se utiliza, en este caso ATmega256RFR2, además todas las extensiones deben estar actualizadas para que no se produzcan errores cuando se compila el código.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Campaña, H. Cabrera, A. Cervelion, y A. Aguirre, *REDES DE SENSORES INALÁMBRICAS (WSN) CASO DE APLICACIÓN*, 1ra ed. Bogotá: Sello Editorial UNAD, 2018.
- [2] A. Cama, E. De la Hoz, y D. Cama, “Las redes de sensores inalámbricos y el Internet de las cosas”, *INGE CUC*, vol. 8, n.º 1, pp. 163-172, oct. 2012.
- [3] N. Aakvaag y J. Frey, “Redes de sensores inalámbricos”, *Revista ABB*, vol. 2, pp. 39-42, ene. 2006.
- [4] V. E. Enríquez, “DISEÑO DE UNA RED INALÁMBRICA PARA UNA WSN DE UN SISTEMA DE ALERTA TEMPRANA DE INCENDIOS PARA EL “BOSQUE PROTECTOR GUAYABILLAS” “, Ibarra: Universidad Técnica del Norte, 2018. [En línea]. Disponible en:
<http://repositorio.utn.edu.ec/bitstream/123456789/8608/1/04%20RED%20203%20TRABAJO%20DE%20GRADO.pdf>. [Accedido: sep. 14, 2021]
- [5] D. M. A. Córdoba y F. A. S. Buitrago, “ESTADO DEL ARTE DE LAS REDES DE SENSORES INALÁMBRICOS”, *TIA*, vol. 2, n.º 1, p. 7, dic. 2013.
- [6] M. C. Acosta, “ESTUDIO DEL ESTÁNDAR IEEE 802.15.4 “ZIGBEE” PARA COMUNICACIONES DE ÁREA PERSONAL DE BAJO CONSUMO DE ENERGÍA Y SU COMPARACIÓN CON EL ESTÁNDAR IEEE 802.15.1 “BLUETOOTH” “, Quito: Escuela Politécnica Nacional, 2006. [En línea]. Disponible en:
<https://bibdigital.epn.edu.ec/bitstream/15000/55/1/CD-0024.pdf>. [Accedido: sep. 15, 2021]
- [7] E. A. Salguero, «IMPLEMENTACIÓN DE UN MEDIDOR DE NIVEL DE POTENCIA PARA EL ESTÁNDAR IEEE 802.15.4.», Quito: Escuela Politécnica Nacional, 2016. [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/15242/1/CD-7016.pdf>. [Accedido: sep. 15, 2021]
- [8] N. Salman, I. Rasool, y A. Kemp, “Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks” in ISWCS, sep. 2010, pp. 701-705.
- [9] Jin-Shyan Lee, “An Experiment on Performance Study of IEEE 802.15.4 Wireless Networks”, in IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy, 2005, vol. 2, pp. 451-458.

- [10] J. T. Adams, «An Introduction to IEEE STD 802.15.4», in IEEE Aerospace Conference, Big Sky, MT, USA, 2006, pp. 1-8.
- [11] F. D. C. Reyes, "IMPLEMENTACIÓN DEL ALGORITMO DE PROTOCOLO DE DIRECCIONAMIENTO PARA REDES DE SENSORES INALÁMBRICOS CON EL ESTANDAR IEEE 802.15.4.", Quito: Escuela Politécnica Nacional, 2018. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/19061>. [Accedido: 16-sep-2021]
- [12] J. D. Díaz y M. F. Mejía, "IMPLEMENTACIÓN DE UN ALGORITMO PARA LA DETECCIÓN DE NODOS CAÍDOS Y FALLOS DE ENLACE EN TOPOLOGÍAS TIPO ÁRBOL UTILIZANDO EL ESTÁNDAR IEEE 802.15.4", Quito: Escuela Politécnica Nacional, 2019. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/20267>. [Accedido: 16-sep-2021]
- [13] P. Dhillon y D. H. Sadawarti, «A Review Paper on Zigbee (IEEE 802.15.4) Standard», International Journal of Engineering Research, vol. 3, n. ° 4, p. 6, abr. 2014.
- [14] A. F. González, "VALIDACIÓN DE UNA RED DE SENSORES INALÁMBRICOS BASADA EN EL ESTÁNDAR IEEE 802.15.4", Bogotá: Universidad de los Andes, 2006. [En línea]. Disponible en: <https://repositorio.uniandes.edu.co/bitstream/handle/1992/23136/u281737.pdf?sequence=1>. [Accedido: 16-sep-2021]
- [15] E. Callaway et al., "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks", IEEE Commun. Mag., vol. 40, n. ° 8, pp. 70-77, ago. 2002.
- [16] N. S. Bhat, "Design and Implementation of IEEE 802.15.4 Mac Protocol on FPGA", International Journal of Computer Applications, p. 5, 2011.
- [17] S. Farahani, "ZigBee and IEEE 802.15.4 Protocol Layers", en ZigBee Wireless Networks and Transceivers, Ed. Burlington: Newnes, 2008, pp. 33-135.
- [18] D. KOŚCIELNIK, "Simulation Study of the Influence of the Hidden and Exposed Stations for the Efficiency of IEEE 802.15.4 LR-WPAN Networks," Wireless Sensor Network, Vol. 2, n. ° 1, 2010, pp. 7-17.

- [19] I. C. Society, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), vol. 2006, no. September. New York, 2006.
- [20] “ZigBit Extension User Guide (USER GUIDE)”, Atmel, p. 21. Disponible en: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42186-ZigBit%20Extension-User-Guide.pdf>. [Accedido: dic. 15, 2021]
- [21] “Microcontroller with Low Power 2.4GHz Transceiver for ZigBee and IEEE 802.15.4”, Atmel, p. 611. Disponible en: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8393-MCU_Wireless-ATmega256RFR2-ATmega128RFR2-ATmega64RFR2_Datasheet.pdf. [Accedido: dic. 17, 2021]
- [22] “Atmel AVR2025: IEEE 802.15.4 MAC Software Package - User Guide”, Atmel, p. 94. Disponible en: https://www.avrfreaks.net/sites/default/files/forum_attachments/AVR2025_User_Guide_v3.1.1.pdf. [Accedido: dic. 20, 2021]
- [23] “Atmel® Studio 7 - Developer Help”. Disponible en: <https://microchipdeveloper.com/atstudio:studio7intro>. [Accedido: dic. 20, 2021].
- [24] “Wireless Composer 7.0”. Disponible en: <https://gallery.microchip.com/packages/96EB07D7-6144-450B-A7CF-4D881B611610/> [Accedido: dic. 20, 2021]
- [25] “PACKET-SNIFFER Calculation tool | TI.com”. Disponible en: <https://www.ti.com/tool/PACKET-SNIFFER>. [Accedido: dic. 20, 2021].

ANEXOS

ANEXO A. Código fuente

ANEXO B. Diagramas de flujo

ANEXO C. Manual de creación y configuración de proyecto en Atmel Studio 7.0

ANEXO D. Manual de grabado del código en los transceptores ATZB-256RFR2-XPRO

ORDEN DE EMPASTADO