

Universidad de Oriente
Facultad de Ingeniería Eléctrica
Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

**Red de sensores inalámbricos para la “Casa
de Cultivos Protegidos San José”**

Autor: Felix Manuel Piña Figueredo

Tutores: M.Sc. Lidice Romero Amondaray

Dra.C. María Margarita Goire Castilla

Santiago de Cuba

Junio, 2015

Universidad de Oriente
Facultad de Ingeniería Eléctrica
Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

**Red de sensores inalámbricos para la “Casa
de Cultivos Protegidos San José”**

Autor: Felix Manuel Piña Figueredo

Correo: felix.pina@tle.fie.uo.edu.cu

Tutores: M.Sc. Lidice Romero Amondaray

Profesor Auxiliar, Departamento de Telecomunicaciones, Facultad de Ingeniería Eléctrica,

Correo: lidice@fie.uo.edu.cu

Dra.C. María Margarita Goire Castilla

Profesor Titular, Departamento de Telecomunicaciones, Facultad de Ingeniería Eléctrica,

Correo: mgoire@fie.uo.edu.cu

Santiago de Cuba

Junio, 2015



COMPROMISO DEL AUTOR

Hago constar que el presente trabajo de diploma es de mi autoría exclusivamente, no constituyendo copia de ningún trabajo realizado anteriormente y las fuentes usadas para la realización del trabajo se encuentran referidas en la bibliografía. Doy mi consentimiento a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización del Tutor o Institución.

Firma del Autor

PENSAMIENTO

“Comienza a manifestarse la madurez cuando sentimos que nuestra preocupación es mayor por los demás que por nosotros mismos”

Albert Einstein

DEDICATORIA

A mis padres queridos, Marbelis y Enrique, que me han apoyado incondicionalmente a lo largo de toda mi vida. Gracias por toda una vida de cariño, educación y comprensión. Ya tienen un hijo ingeniero.

AGRADECIMIENTOS

A mis padres y hermana por darme su apoyo incondicional y hacer este sueño realidad. Sin ustedes nada de esto hubiese sido posible. Gracias por existir.

Aunque resulte repetitivo, debo destacar a alguien que hizo algo extraordinario por mí, mi padre Enrique, muchas gracias por tu guía y todo tu esfuerzo.

A todos mis familiares, en especial a mis abuelos Maira, Juan y Miriam, por su apoyo durante todos estos años.

A la pequeña semilla que va creciendo en el vientre de mi madre, la cual me dio fuerzas para seguir adelante cuando creía que era imposible.

A mis tutoras, las profesoras Lidice Romero y Margarita Goire, por el sacrificio y la dedicación con que trabajaron en esta investigación.

A mi gran amigo Mario, por su amistad incondicional y su ayuda en todo momento.

A mis amigos Tony, Daniel, Eloy, Torres, Yordan, el Lucky, por estar siempre ahí, en las buenas y en las malas, y ayudarme en los momentos más difíciles.

Al resto de mis compañeros por su confianza y por haber estado presente estos cinco años.

A todos aquellos que de una forma u otra ayudaron a cumplir este sueño.

RESUMEN

La efectividad de los cultivos tapados depende del conocimiento del estado de los cultivos y del sensado de los parámetros que los afectan, aspectos resueltos por la llamada agricultura de precisión, a través de modernas tecnologías como son las inalámbricas. La “Casa de Cultivos Protegidos San José” de Santiago de Cuba no cuenta con un sistema de supervisión que permita medir la temperatura, la humedad relativa y la radiación solar, lo que compromete su rendimiento. El resultado obtenido en esta tesis da respuesta a esta problemática con una propuesta de red inalámbrica para la medición de los parámetros mencionados. La red que se diseñó cuenta con un total de 25 nodos distribuidos en 23 casas de cultivo, una de posturas y un nodo coordinador ubicado en la estación base. Se simularon las topologías malla y *cluster-tree*, utilizando el NS2 (*Network Simulator 2*), y se comprobó que esta última es la adecuada para ser utilizada en el diseño propuesto, ya que mejora la razón de entrega de paquetes y, particularmente, tiene un bajo consumo energético. Para el análisis de los resultados se programó un filtro con el lenguaje AWK, debido a que la herramienta que posee el NS2 no es compatible con las trazas generadas en las simulaciones de redes de sensores inalámbricos.

Palabras clave: Agricultura de precisión, NS2, topologías, AWK.

ABSTRACT

The effectiveness of the crops covered depends on knowledge of the state of crops and the census of the parameters that affect it, aspects solved by the so-called precision agriculture through modern technologies such as wireless. The "Protected Crops House San José" in Santiago de Cuba does not have a monitoring system to measure temperature, relative humidity and solar radiation, compromising performance. The result obtained in this thesis is addressing this issue with a proposed wireless network for measuring these parameters. The network was designed with a total of 25 nodes distributed in 23 houses of crops, one of posture and a coordinator node located in the base station. Mesh topologies and cluster-tree were simulated using the NS2 (Network Simulator 2), and found that the latter is suitable for use in the proposed design, as it improves packet delivery ratio and, in particular, has low power consumption. For analysis of the results a filter with AWK language was planned, because the tool has the NS2 is not compatible with the traces generated in simulations of wireless sensor networks.

Keywords: *precision agriculture, NS2, topology, AWK.*

ÍNDICE

INTRODUCCIÓN.....	1
CAPITULO 1 . REDES DE SENSORES INALÁMBRICOS.....	5
1.1 Introducción a las redes de sensores inalámbricos.....	5
1.2 Topologías de las redes de sensores inalámbricos	7
1.3 Dispositivos en las redes de sensores inalámbricos	9
1.3.1 <i>Hardware</i> de un nodo sensor	9
1.3.2 <i>Software</i> de un nodo sensor	10
1.4 Estándares de comunicación	11
1.4.1 <i>Bluetooth</i>	11
1.4.2 Wi-Fi.....	12
1.4.3 <i>ZigBee</i>	13
1.5 Protocolos de enrutamiento en redes de sensores inalámbricos.....	19
1.5.1 DSDV.....	20
1.5.2 DSR.....	20
1.5.3 AODV	21
1.6 Sistemas operativos para redes de sensores inalámbricos.....	21
1.6.1 TinyOS.....	21
1.6.2 Contiki	22
1.6.3 MANTIS	22
1.7 Simuladores para redes de sensores inalámbricos	23
1.7.1 NS2	23
1.7.2 Omnet ++	23
1.7.3 J-Sim	24
1.8 Aplicaciones de las redes de sensores inalámbricos	24
1.8.1 Aplicaciones en la agricultura.....	24
1.8.2 Otras aplicaciones	26
CAPITULO 2 . PLATAFORMAS DE REDES DE SENSORES INALÁMBRICOS	28
2.1 Plataforma MICAz.....	28
2.2 Plataforma Mica2	30
2.3 Plataforma Mica2Dot.....	31

2.4	Plataforma Imote2.....	32
2.5	Plataforma Iris.....	33
2.6	Plataforma <i>Tmote Sky</i>	34
2.7	Plataforma <i>eKo</i>	34
2.8	Plataforma <i>Waspote</i>	36
CAPITULO 3 . DISEÑO DE LA DE RED DE SENSORES INALÁMBRICOS PARA LA		
“CASA DE CULTIVOS PROTEGIDOS SAN JOSÉ”		40
3.1	“Casa de Cultivos Protegidos San José”	40
3.2	Control climático en casas de cultivo.....	42
3.2.1	Temperatura.....	43
3.2.2	Humedad relativa.....	43
3.2.3	Iluminación	44
3.3	Diseño de la red de sensores inalámbricos para la “Casa de Cultivos Protegidos	
	San José”	44
3.3.1	Especificaciones de diseño	45
3.4	Simulación de las topologías de red.....	46
3.4.1	<i>Network Simulator 2 (NS2)</i>	46
3.4.2	Estructura de simulación.....	47
3.4.3	Configuración de la simulación	49
3.4.4	Simulación de la topología malla.....	50
3.4.5	Simulación de la topología <i>cluster-tree</i>	51
3.5	Valoración económica.....	54
CONCLUSIONES Y RECOMENDACIONES		57
GLOSARIO DE TÉRMINOS		61
ANEXOS		64

INTRODUCCIÓN

Las redes de sensores inalámbricos son cada vez más populares, especialmente en aplicaciones que requieren un monitoreo continuo de diversas variables del entorno, en las que el uso de sensores tradicionales es inviable. Dominios de aplicación tan variados como monitorización ambiental, industrial, maniobras militares o agricultura de precisión, son algunos de los ejemplos para los que esta tecnología es de gran ayuda.

Una red de sensores inalámbricos (en inglés: *Wireless Sensor Network* o *WSN*) es una red de diminutos dispositivos, equipados con sensores, que colaboran en una tarea común y están distribuidos en un área geográfica determinada. Estos dispositivos, conocidos como nodos sensores o motes, son unidades autónomas capaces de recopilar información, procesarla y comunicarse con otros nodos conectados a la red [1].

Las redes de sensores se caracterizan por ser redes desatendidas, sin intervención humana, habitualmente construidas para ejecutar una única aplicación. A través de estas redes inteligentes se pueden integrar funcionalidades que antes eran independientes unas de otras, con el fin de lograr una máxima eficiencia. Este tipo de redes permiten formar redes *ad-hoc* sin infraestructura física preestablecida ni administración central [2].

La agricultura juega un papel crucial en el desarrollo de la civilización humana. El incremento de la demanda de alimentos a nivel mundial en los últimos años provoca una tendencia a aplicar esfuerzos extras y técnicas especiales para multiplicar la producción de nutrientes. El uso de diferentes tecnologías de la información dirigidas a este sector es un ejemplo de estos empeños.

En la agricultura, las redes de sensores inalámbricos se utilizan para aumentar la eficiencia en la producción y el crecimiento de la cosecha. Por lo general, los datos de los sensores corresponden a las condiciones ambientales, tales como la temperatura, velocidad del viento, la humedad relativa y las propiedades físicas y químicas del suelo [3] [4].

En este sentido se puede mencionar el proyecto *WiseVine* [5], en el cual se implanta una red de 66 nodos en un viñedo, que proporciona interesantes datos para los ingenieros

agrónomos encargados del análisis de los resultados. En otros proyectos [6] [7], se implementan sistemas de adquisición de datos basados en WSN, con el objetivo de obtener mediciones de temperatura, humedad relativa y luz en casas de cultivo.

Las casas de cultivos son instalaciones que tienen como objetivo proteger a las plantas de la incidencia de la intensa radiación solar y de las fuertes lluvias propias de los países tropicales. Las mismas permiten modificar, total o parcialmente las condiciones ambientales, para que las plantas se desarrollen en un medio más favorable que el existente al aire libre [8].

El auge de las casas de cultivo en Cuba se inicia a partir de la transferencia de tecnologías de otros países, principalmente Israel y España. Sin embargo, teniendo en cuenta las características del clima, a los diseños originales se les incluyen modificaciones, buscando el llamado efecto sombrilla, de manera que puedan producirse hortalizas durante todo el año, y fundamentalmente en verano, en momentos en que resulta imposible lograrlo a cielo abierto [9].

Hoy, Cuba lleva a cabo un intenso programa de desarrollo de cultivos protegidos, al que se destinan considerables recursos. Los objetivos primordiales son recuperar las capacidades existentes, y crear nuevas áreas, dotadas de la tecnología adecuada y que respondan a la demanda actual de alimentos [10].

La “Casa de Cultivos Protegidos San José”, ubicada en la provincia Santiago de Cuba, posee un área total de 1,8 hectáreas y está conformada por 23 casas de cultivo y una casa de posturas. Entre las especies que se cosechan están el tomate, pepino, pimiento, sandía, melón, col, lechuga, entre otros; y su destino principal es el turismo, con lo cual se sustituyen costosas importaciones [11].

La calidad y rendimiento de los cultivos que se desarrollan en estas casas depende de las condiciones ambientales que se logren mantener en su interior. Las etapas de desarrollo de los cultivos están condicionadas por tres factores principales: radiación solar, temperatura y humedad relativa. Las casas de cultivo deben mantener estas variables dentro de ciertos intervalos para mantener el metabolismo de las especies que se cosechan [12].

Los trabajadores del centro no realizan mediciones de los parámetros antes mencionados, debido a que no cuentan con instrumentos de medición que les posibilite realizar estas observaciones. Otro factor importante es el sistema de riego. La irrigación se realiza

diariamente, entre las siete y las ocho de la mañana, y la duración depende de la fase fenológica en la que se encuentre el cultivo. Normalmente se ejecuta solo una vez al día, excepto en las fechas de mucha radiación solar, en las que se lleva a cabo en la mañana y en la tarde. La decisión de seleccionar una u otra alternativa se toma de forma empírica, basada en la experiencia de los obreros [13].

Problema de investigación:

La “Casa de Cultivos Protegidos San José” no cuenta con un sistema de supervisión que permita medir la temperatura, la humedad relativa y la radiación solar, lo que afecta el rendimiento de las especies cultivadas.

Objeto de estudio:

Redes de sensores inalámbricos.

Objetivo general:

Diseñar una red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José” que permita realizar mediciones de temperatura, humedad y radiación solar.

Objetivos específicos:

- Realizar un estudio del estado del arte de las redes de sensores inalámbricos.
- Comparar algunas de las plataformas disponibles en el mercado para soluciones de redes de sensores inalámbricos.
- Realizar un diseño de una red de sensores inalámbricos que permita medir temperatura, humedad relativa y radiación solar en la “Casa de Cultivos Protegidos San José”.

Tareas

- Estudio de las características generales de las redes de sensores inalámbricos.
- Comparación de los diferentes estándares de comunicación utilizados en redes de sensores inalámbricos.
- Estudio de las características del estándar de comunicación *ZigBee*.
- Estudio del estado de esta técnica en el ámbito de la agricultura.
- Comparación de algunas de las plataformas disponibles para soluciones de redes de sensores inalámbricos.
- Diseño de la solución de redes de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”.

- Evaluación de la topología de red más adecuada para la “Casa de Cultivos Protegidos San José”.
- Redacción del informe de la tesis

Estructura de la tesis

La tesis está estructurada en resumen e introducción, tres capítulos, conclusiones, recomendaciones y tres anexos.

En el capítulo 1 se presenta un estudio del estado del arte de las redes de sensores inalámbricos. Se realiza una descripción del estándar de comunicación *ZigBee*, y se revisa el estado de esta técnica en el ámbito de la agricultura.

En el capítulo 2 se realiza una comparación de algunas de las principales plataformas de redes de sensores inalámbricos disponibles en el mercado. Se analizan sus características y su posible aplicación a entornos agrícolas.

En el capítulo 3 se presenta el diseño de la red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”. Se realiza una comparación entre las topologías malla y *cluster-tree*, utilizando el simulador NS2 (*Network Simulator 2*) y se programa un filtro en el lenguaje AWK para el análisis de los resultados.

CAPITULO 1 . REDES DE SENSORES INALÁMBRICOS

Una red de sensores inalámbricos es un conjunto de dispositivos electrónicos con tamaño reducido, elevada autonomía y baja capacidad de cómputo, que cooperan entre sí, vía radio, para llevar a cabo una tarea común.

En este capítulo se presenta una revisión del estado del arte de las redes de sensores inalámbricos. Para ello, en primer lugar se realiza una descripción general, y posteriormente, se describen con mayor detalle las características de esta tecnología tales como: las topologías de red, protocolos de comunicación, así como una revisión del estado de esta técnica en el ámbito de la agricultura.

1.1 Introducción a las redes de sensores inalámbricos

La recolección de datos es una tarea indispensable en un gran número de aplicaciones. Ejemplos de las mismas son las ambientales, para el estudio de las condiciones de un terreno o las militares para el control de equipamiento o tropas. En el pasado, tanto operadores humanos de forma manual, como equipos sensores conectados a una central mediante enlaces cableados eran los actores dedicados a la recogida de datos.

Las dos opciones presentan problemas: las mediciones realizadas por operadores humanos son costosas en tiempo, lo que restringe la cantidad de ubicaciones en la que realizar medidas. Por otro lado, el uso de sensores cableados obliga al despliegue de grandes cantidades de cable e impide las observaciones en lugares de difícil acceso.

Con los avances tecnológicos en los campos de la electrónica y las comunicaciones inalámbricas se resuelven estos problemas, abaratando el uso de dispositivos inalámbricos para la adquisición de datos, lo que permite el auge de las redes de sensores.

La tecnología de redes de sensores inalámbricos constituye una elección muy oportuna para los casos en los que se deben recoger datos de múltiples ubicaciones. Su uso permite el despliegue de un gran número de dispositivos sensores de bajo costo que forman una red inalámbrica robusta, escalable y adaptable a los cambios en el entorno o en su topología.

Una red de sensores inalámbricos se define como un conjunto de nodos sensores, que consisten en pequeños sistemas informáticos embebidos, encargados de recoger datos del entorno como temperatura, humedad o presión atmosférica, y que pueden comunicarse de forma inalámbrica [2].

Los datos de la red son recopilados por un nodo conocido como estación base o sumidero, que se encarga de recibir la información de todos los nodos de la red. Este nodo central suele estar conectado a un ordenador o a un dispositivo capaz de almacenar gran cantidad de información. Una vez que los datos son recogidos por los sensores, son procesados por los nodos y enviados a la estación base utilizando diversos protocolos de comunicación inalámbricos. En la figura 1.1, se muestra la infraestructura de una red inalámbrica de sensores.

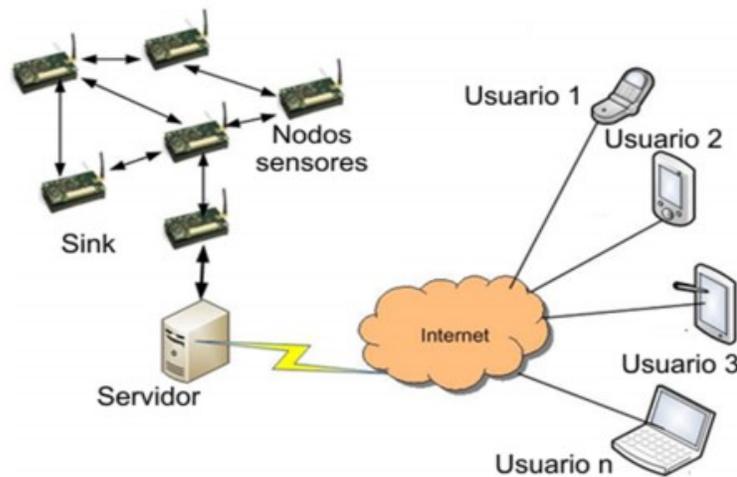


Figura 1.1. Infraestructura de una red de sensores inalámbricos (Fuente: [1]).

La configuración de la red puede ser tan variada como los campos de aplicación. Un nodo fuente es una entidad en la red que puede proveer información. Por otro lado, un sumidero es la entidad que requiere la información. El sumidero puede pertenecer a la red de sensores (y ser otro sensor), ser una entidad externa a la red o ser un *gateway* a otra red más grande, como internet [2]. El modelo de red típico comprende un sumidero y múltiples nodos fuente, pero en muchas aplicaciones se utilizan también múltiples sumideros.

Las WSN pueden ser utilizadas en casi cualquier ámbito donde se necesite una monitorización continua, por lo que el despliegue de los nodos puede no seguir ningún tipo de orden. Los algoritmos y protocolos utilizados deben ser capaces de funcionar de forma

autónoma y adaptable con el fin de satisfacer eficientemente los requerimientos de cada aplicación.

Debido a la naturaleza de los nodos y a las diversas aplicaciones en las que pueden ser utilizadas, las WSN tienen características que las diferencian del resto de las redes inalámbricas [14] [15]. Estas características son:

- **Limitaciones:** el pequeño tamaño de los nodos no permite, al menos por el momento, la incorporación en ellos de potentes procesadores ni de sistemas de almacenamiento de gran capacidad. Además, la energía, proporcionada por baterías, limita el tiempo de funcionamiento de los nodos. Por esto, se presentan restricciones en cuanto a procesamiento, almacenamiento, energía y comunicaciones.
- **Escalabilidad:** el número de nodos desplegados para cumplir cierta tarea puede ser mucho mayor que los empleados en las redes de área local tradicionales. Así, las técnicas de comunicación empleadas deben mantener su funcionalidad y eficiencia cuando crezca el número de nodos de la red.
- **Auto-configuración:** la configuración manual de cada uno de los nodos se torna imposible al manejar redes muy densas. Por ello, las redes de sensores deben ser capaces de operar de forma autónoma y desatendida, adaptándose a posibles cambios que se produzcan debido a la incorporación, eliminación y cambios de localización de los nodos.
- **Simplicidad:** como consecuencia de las limitaciones de los nodos y el tamaño de las redes, las aplicaciones y protocolos desarrollados deben ser lo más simple posible.
- **Especificidad:** la gran cantidad de parámetros y opciones disponibles a la hora de diseñar una red de sensores hacen que sean altamente dependientes de la aplicación, por lo que muchas de las propuestas presentes en la literatura están orientadas a determinados tipos de aplicación.

1.2 Topologías de las redes de sensores inalámbricos

Dentro del área de sensado, los sensores se interconectan por medio de enlaces inalámbricos, de corta distancia y baja potencia de transmisión, para enviar información a estaciones recolectoras o de monitoreo. Típicamente se despliegan en grandes cantidades y con una distribución densa. Existen dos tipos de redes [16]:

- **No estructuradas:** comprende una colección de nodos densa, desplegados *ad-hoc*, posiblemente al azar. Una vez desplegados, la red opera desatendida, monitoreando y reportando información. El mantenimiento, la administración de la conectividad y detección de fallas son difíciles por la gran cantidad de nodos.
- **Estructuradas:** todos o algunos de los nodos son desplegados de manera pre-planificada, colocados en posiciones fijas. Este tipo de redes tienen la ventaja de requerir una menor cantidad de nodos para lograr la cobertura del área, con un menor costo de administración y mantenimiento.

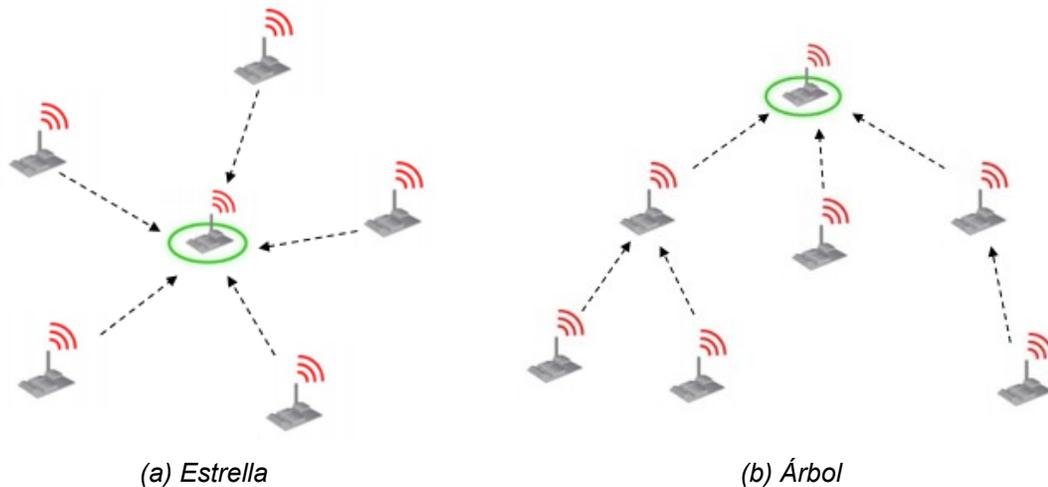


Figura 1.2. Topologías en estrella y árbol.

La estructuración lógica de la red la llevan a cabo los protocolos de red y de acceso al medio. Al carecer de infraestructuras físicas, las redes de sensores pueden adoptar la estructura que más convenga en cada aplicación. Las topologías más comunes de las redes de sensores son malla, estrella, árbol y basadas en *cluster* [17].

- **Topologías en estrella:** son sistemas de un solo salto, en el que todos los nodos pueden comunicarse de forma directa con la estación base (ver figura 1.2(a)). Los nodos no pueden comunicarse directamente con el resto; todas las comunicaciones tienen que pasar por la estación base. Una analogía al mundo de las redes en general es la arquitectura cliente-servidor de las redes Ethernet.
- **Topologías en árbol:** pueden ser consideradas como una generalización de las redes en estrella. Los caminos definidos en estas redes van desde cualquier nodo hasta la estación base, pudiendo atravesar varios nodos intermedios hasta llegar a ella (ver figura 2(b)).

- **Topologías en malla:** estas redes permiten a los datos saltar de nodo en nodo. De esta forma, un nodo es capaz de comunicarse con cualquier otro, de modo que los datos pueden atravesar nodos intermedios hasta llegar a su destino (ver figura 1.3(a)).
- **Topologías basadas en *cluster*:** esta clase de topología separa los nodos en grupos gobernados por un cabeza de *cluster* (ver figura 3(b)). Los cabeza de grupo pueden comunicarse entre sí, de forma que un nodo que quiera enviar datos a otro, se los envía a su cabeza de *cluster* y este lo reenvía al nodo destino, si pertenece a su grupo, o a un cabeza de *cluster* vecino que conozca la ruta hacia el destino.

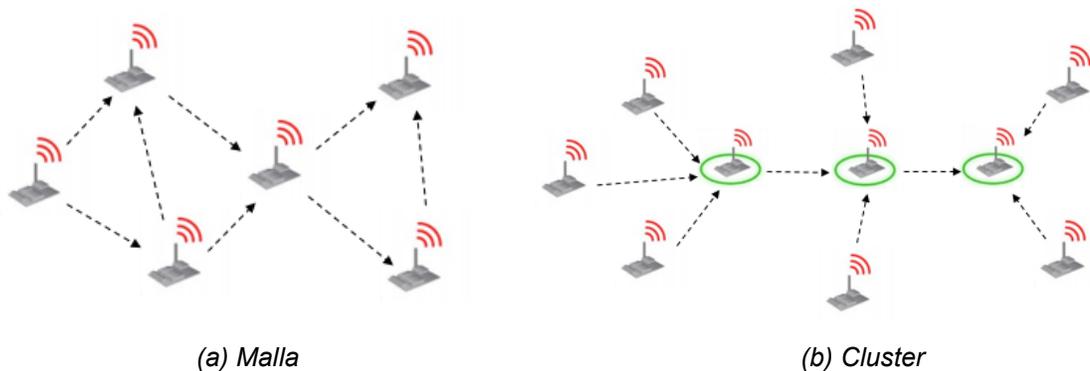


Figura 1.3. Topologías en malla y cluster.

1.3 Dispositivos en las redes de sensores inalámbricos

Los nodos son la pieza básica de una red de sensores inalámbricos. Se tratan de dispositivos capaces de realizar medidas acerca de un determinado fenómeno y propagarlos vía radio. Para lograr esto, los nodos están formados por subsistemas de *hardware* y de *software*. A continuación se explica cada uno de ellos y qué elementos lo componen.

1.3.1 *Hardware* de un nodo sensor

Un nodo está formado por los subsistemas de *hardware* que se muestran en la figura 1.4. Existen cuatro subsistemas que siempre se encuentran presentes en un nodo. El primero de ellos, la unidad de procesamiento, se encarga de coordinar al resto de los subsistemas, disparando las tareas de recogida de datos y la comunicación con otros nodos, así como procesar la información recolectada. La unidad de sensado toma los datos del entorno a través de sus sensores y transforma las medidas analógicas adquiridas en medidas digitales que el microprocesador pueda manejar [18].

Por su parte, el transceptor es el responsable de la comunicación con los otros nodos conectados a la red. Por último, se encuentra la unidad de alimentación, que suministra la energía al resto de los subsistemas. Además existen otros subsistemas que pueden encontrarse opcionalmente, como podrían ser movilizadores (si el nodo necesita desplazarse), subsistemas de localización, o un generador de potencia (por ejemplo, un panel fotovoltaico).

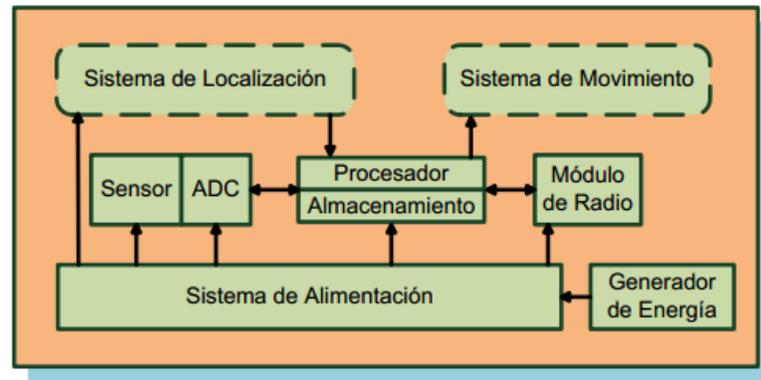


Figura 1.4. Componentes de un nodo sensor (Fuente: [18]).

Los motes son dispositivos limitados en cuanto a sus prestaciones. Su CPU (*Central Processing Unit*) funciona a baja frecuencia, disponen de pocos Kbytes de memoria RAM (*Random Access Memory*) y tienen una fuente de alimentación que deben conservar durante el mayor tiempo posible. Debido a esto, las funciones que realizan los nodos se limitan a la captura y envío de datos, con algunas tareas de procesado destinadas a disminuir la cantidad de información que se transmite.

El almacenamiento en memoria no volátil está menos restringido, ya que disponen de una memoria *flash* de cientos de Kbytes, lo que permite el almacenamiento de gran cantidad de datos, que no tienen por qué ser transmitidos inmediatamente. Además, los nodos cuentan con puertos de entrada/salida que permiten tanto la conexión de subsistemas adicionales, como la comunicación con una computadora o la programación de los mismos.

1.3.2 Software de un nodo sensor

En cuanto a los subsistemas de *software*, los nodos operan básicamente con cinco de ellos, estos son [19]:

- **Sistema operativo:** El micro-código o sistema operativo, también llamado *middleware*, es el micro-código común del dispositivo, utilizado por todos los programas de alto nivel residentes en el nodo. Usualmente, presentan una

arquitectura que permite una rápida implementación con un tamaño mínimo de código. TinyOS es un ejemplo.

- **Controladores de sensores:** Módulos de *software* que administran funciones básicas de los transceptores de sensores.
- **Procesadores de comunicación:** Módulos que administran funciones de comunicación, encaminamiento, almacenamiento y reenvío de paquetes, mantenimiento de la topología de red, control de acceso al medio, cifrado, corrección de errores, entre otros.
- **Controladores de comunicación:** Módulos que administran las tareas de utilización del enlace de transmisión por radio, sincronización, codificación de señales, recuperación de errores, modulación, entre otros.
- **Mini-aplicaciones de proceso de datos:** Aplicaciones básicas soportadas a nivel de nodo para procesamiento de datos en la red.

1.4 Estándares de comunicación

Una WSN debe funcionar de manera autónoma, aunque ocurran situaciones como el fallo de uno o varios nodos sensores o la adición de nuevos nodos a la red. Además, en determinadas aplicaciones, los motes pueden estar desplegados en lugares de difícil acceso. En esta línea, es necesario utilizar estándares y protocolos de comunicación que garanticen lo anteriormente expuesto.

1.4.1 Bluetooth

Bluetooth es una tecnología utilizada para la conectividad inalámbrica de corto alcance entre dispositivos como PDA (*Personal Digital Assistance*), teléfonos celulares, teclados, impresoras, entre otros. Es una tecnología de radiofrecuencia que trabaja en la banda de 2.4 GHz y utiliza salto de frecuencia para expansión del espectro. La distancia de conexión puede ser de hasta 10 metros o más, en dependencia del incremento de la potencia del transmisor [20].

Se desarrollan prototipos de redes de sensores basadas en este estándar. Los nodos deben estar organizados en pico redes con un nodo maestro y un máximo de siete nodos esclavos activos. La aplicación de esta tecnología a las redes de sensores inalámbricos presenta los siguientes inconvenientes [16]:

- la necesidad de tener un nodo maestro constantemente, con el costo de interrogar sus esclavos,
- la cantidad limitada de esclavos por pico red que soporta,
- para el caso de redes de sensores densas, se necesita un número enorme de nodos maestros,
- un esclavo activo debe permanecer siempre encendido, ya que no puede predecir cuándo será interrogado por el maestro,
- un esclavo pasivo debe postularse con el maestro para cambiar a activo, y si ya hay siete nodos activos, es rechazado,
- se requiere que cada nodo pueda asumir el rol de maestro o esclavo, lo que agrega una complejidad considerable,
- los rápidos saltos de frecuencia requieren una sincronización estricta entre los nodos de la pico red.

1.4.2 Wi-Fi

Wi-Fi es una de las tecnologías más usadas en la actualidad debido a su bajo costo y compatibilidad con la mayoría de los dispositivos inalámbricos. La misma reúne un conjunto de estándares de redes inalámbricas basadas en las normas de la IEEE 802.11. Las velocidades de transferencia de datos se encuentran entre los 11 Mbps y los 54 Mbps. Permite trabajar con porciones del espectro de bandas no licenciadas como es el caso de la banda ISM (*Industrial, scientific and medical*) en los 2.4 GHz y de la banda UNII (*Unlicensed National Information Infrastructure*) en los 5 GHz. El método de acceso al medio utilizado es CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) [21]. Su alcance de radio está entre los 30 y 100 metros. En términos generales presenta los siguientes inconvenientes [16]:

- requiere que los nodos estén permanentemente escuchando el medio, ya que podrían tener que recibir una trama en cualquier momento,
- si bien se proveen algunas funcionalidades de ahorro de energía, en general está orientado a altas tasas de transmisión, y los transceptores disponibles requieren una cantidad de energía mucho mayor que lo aceptable en aplicaciones de redes de sensores,
- es un protocolo de salto único para redes *ad-hoc*, cuando lo común en redes de sensores es el encaminamiento de salto múltiple.

1.4.3 ZigBee

Los estándares analizados anteriormente están orientados hacia aplicaciones con altos requerimientos de ancho de banda, pero ninguno de estos se adecúa para aplicaciones que requieran una baja tasa de transmisión de datos y bajos requerimientos de energía. Por estas razones, surge la necesidad de crear un nuevo estándar inalámbrico que asociara autonomía, envío de datos de baja capacidad y bajo costo.

Bajo estas premisas aparece *ZigBee*, tecnología que trata de dar respuesta a las aplicaciones en las que el costo y el consumo de potencia son parámetros críticos de diseño. El mismo está basado en la norma IEEE 802.15.4, que implementa las capas físicas y de control de acceso al medio, mientras que las capas de red y de aplicación son establecidas por la Alianza *ZigBee* [22]. La misma está enmarcada dentro de un subgrupo de las redes de área personal, conocido como LR-WPAN (*Low Rate Wireless Personal Area Network*). En la figura 1.5 se muestran las capas que conforman el estándar.

Dentro de las características más importantes se puede destacar que alcanza tasas de transmisión de hasta 250 Kbps y tiene un rango de cobertura de entre 10 y 75 metros, operando en las bandas ISM a las frecuencias de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos). Seguidamente se presenta una breve descripción de cada una de las capas que conforman el estándar.

Capa física

La capa física define las frecuencias de radio que se utilizan para los enlaces y los mecanismos de control de bajo nivel. La misma constituye la más baja del protocolo y es la más cercana al *hardware*, y se comunica y controla directamente al transceptor de radio. Es responsable de la activación del equipo que transmite o recibe los paquetes. Algunas características globales de esta capa de la arquitectura son el control del transceptor de radio, calidad del enlace, selección del canal, detección de los niveles de energía recibidos, entre otros.

El estándar IEEE 802.15.4 ofrece dos tipos de capa física, que se combina con la capa MAC (*Medium Access Control*) para permitir un amplio rango de aplicaciones en la red. La principal diferencia entre ambas capas radica en las bandas de frecuencia [22]. En la figura 1.6 se observa la distribución de los canales de radio.

La disponibilidad internacional de la banda de los 2.4 GHz ofrece ventajas en términos de mercados más amplios y costos de manufactura más bajos. Por otro lado, las bandas de

868 y 915 MHz ofrecen una alternativa a la congestión creciente y demás interferencias (hornos de microondas, teléfonos inalámbricos, entre otros) asociados a la banda de los 2.4 GHz y mayores rangos por enlace, debido a que existen menores pérdidas de propagación.

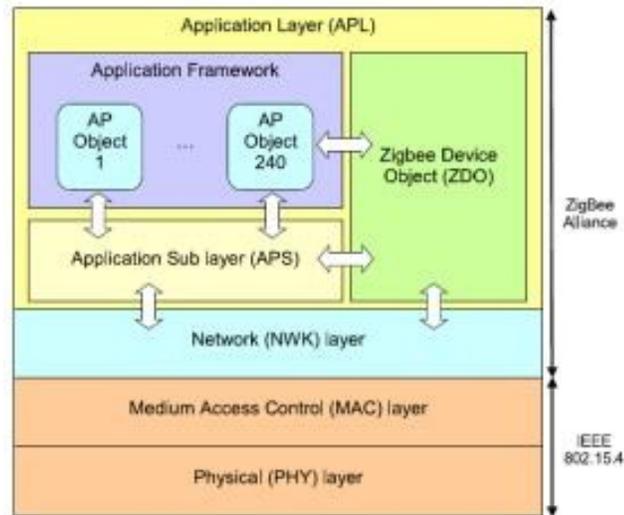


Figura 1.5. Arquitectura ZigBee (Fuente: [22]).

Una segunda distinción de las características de la capa física es la tasa de transmisión de datos. La capa física de 2.4 GHz permite una tasa de transmisión de 250 Kbps, mientras que la capa física de los 868/915 MHz ofrece una tasa de 20 Kbps y 40 Kbps respectivamente.

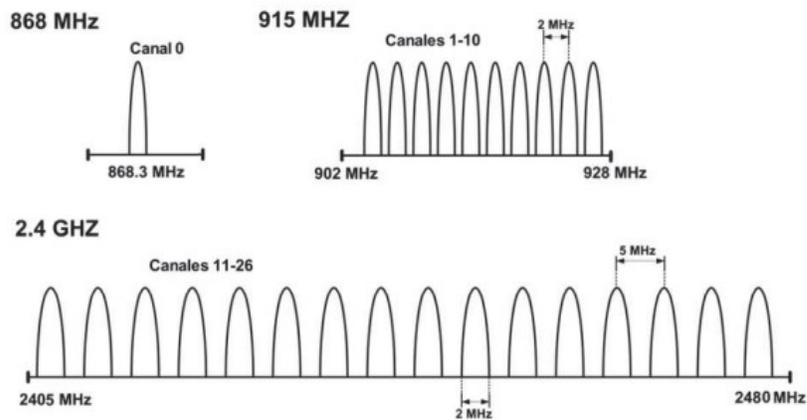


Figura 1.6. Bandas de frecuencia de operación en la capa física (Fuente: [23]).

Capa de acceso al medio (MAC)

La capa de acceso al medio provee una interfaz entre la capa física y la capa de red. La misma es responsable del direccionamiento de los datos salientes y la identificación de la

fuente de los datos entrantes. Se encarga del ensamblaje de los paquetes de datos (tramas) que se transmiten y el desensamblaje de los paquetes de datos recibidos.

La capa MAC se encarga del control de acceso al medio y es responsable de transmitir los paquetes. También lleva las labores de validar las tramas que recibe, comprobar errores en la transmisión y confirmar la recepción de tramas al emisor. Otras funciones importantes que cumple son: la fragmentación de paquetes, control de flujo, tasa de transmisión, funciones relacionadas con la gestión de la batería, generar los *beacons* (guías de red) en el caso de ser un dispositivo coordinador, asociación y disociación, entre otras [22]. En resumen se encarga de controlar un medio de comunicación compartido por una serie de dispositivos que se comunican a través de él.

El estándar IEEE 802.15.4 define dos modos de operación: modo *beacon* habilitado (modo ranurado) y modo *beacon* no habilitado (modo no ranurado). En el caso del primero, el coordinador de la red utiliza una estructura de supertrama para manejar la comunicación entre los dispositivos. El formato de la supertrama lo define el coordinador de la red PAN (*Personal Area Network*). En el caso del modo no ranurado, los nodos envían sus datos empleando el mecanismo CSMA/CA no ranurado. En este modo no se utilizan supertramas. La estructura de la supertrama se muestra en la figura 1.7.

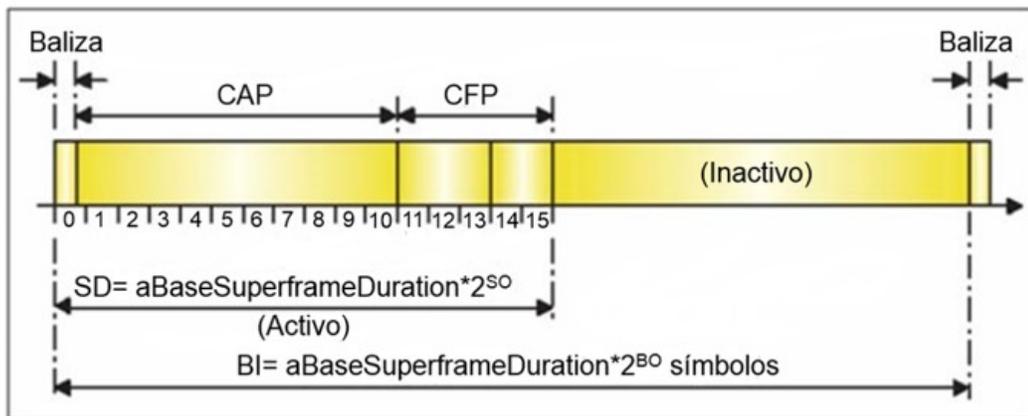


Figura 1.7. Estructura de la supertrama (Fuente: [23]).

La supertrama está dividida en dos períodos: período activo e inactivo. En el caso del período activo está dividido a su vez en dos fases: la fase CAP (*Contention Access Period*), en la que el acceso es por contención y la fase CFP (*Contention Free Period*) en la que el acceso es libre de contención. La porción activa, independientemente de la duración de la supertrama, está dividida en 16 ranuras de igual duración, la cual puede ser configurada.

En la figura 1.7:

- *macBeaconOrder* (BO): Este parámetro define el intervalo en el cual el coordinador debe enviar las tramas *beacons*. Dicho intervalo se denomina BI (*Beacon Interval*). El estándar IEEE 802.15.4 define que $0 \leq BO \leq 14$.

$$BI = aBaseSuperframeDuration * 2^{BO} \text{ símbolos} \quad (1.1)$$

- *macSuperframeOrder* (SO): Este parámetro define el largo de la porción activa de la supertrama. La duración de este fragmento se denomina SD (*Superframe Duration*). El estándar IEEE 802.15.4 define que $0 \leq SO \leq BO$.

$$SD = aBaseSuperframeDuration * 2^{SO} \text{ símbolos} \quad (1.2)$$

El estándar IEEE 802.15.4 define que:

$$aBaseSuperframeDuration = sBaseSlotDuration * aNumSuperframeSlots \quad (1.3)$$

En donde:

$$aBaseSlotDuration = 60 \text{ símbolos} \quad (1.4)$$

$$aNumSuperframesSlots = 16 \text{ slots} \quad (1.5)$$

Capa de red (NWK)

La capa de red constituye la interfaz entre la capa MAC y la capa de aplicación. La misma es responsable de la inicialización de la red y la configuración de entrada y salida a la misma. Se encarga de distribuir las direcciones, desde el coordinador, a los dispositivos que integran la red. Su principal función es el direccionamiento de las tramas de datos hacia el destino. Además, proporciona los medios para garantizar la entrega del paquete al destinatario final, filtra los paquetes recibidos, los cifra y los autentifica.

La capa analizada soporta tres topologías de red: estrella, *cluster-tree* y malla. En el caso de la topología en estrella uno de los dispositivos asume el papel de coordinador de la red. El mismo es responsable de iniciar y mantener los dispositivos. Todos los demás nodos son conocidos como nodos finales y se comunican directamente con el coordinador.

En la topología malla se permiten continuas conexiones y re-configuraciones, saltando de un nodo a otro hasta alcanzar el destino final. Por otro lado, en la topología en árbol un coordinador establece inicialmente la red, los enrutadores forman las ramas al encargarse de la entrega de los mensajes, y los dispositivos finales constituyen las hojas [22].

Capa de aplicación (APL)

La capa de aplicación es la más alta de la arquitectura *ZigBee*. Esta capa provee las funciones que se quieren implementar en los dispositivos. La misma está constituida por tres secciones principales: APS (*Application Support Sub layer*), ZDO (*ZigBee Device Object*) y la estructura de aplicación *ZigBee*.

La APS es responsable de mantener tablas para ligar y enviar mensajes entre dispositivos asociados. La misma es la encargada del proceso conocido como *binding*, que no es más que la habilidad de unir dos dispositivos basados en sus servicios y necesidades. Es además responsable de descubrir otros dispositivos que operan en el área local [20].

Los objetos de dispositivos *ZigBee* proveen una interfaz entre la subcapa APS y la estructura de aplicación. El ZDO tiene la responsabilidad de inicializar la APS, la capa de red y el proveedor de servicio de seguridad. El mismo es el encargado además de configurar el dispositivo como uno de los tres posibles tipos lógicos de dispositivo: dispositivo final, *router* o coordinador [22].

La estructura de aplicación es el ambiente en el cual los objetos de aplicación están organizados para controlar y gestionar las capas del protocolo en un dispositivo *ZigBee*. Los objetos de aplicación son desarrollados por los fabricantes y es aquí donde el dispositivo es personalizado para varias aplicaciones, donde pueden existir 240 objetos de aplicación para cada dispositivo.

Dispositivos *ZigBee*

En una red IEEE 802.15.4 existen dos tipos de dispositivos: dispositivos de funcionalidad completa o FFD (*Full Function Device*), y dispositivos de funcionalidad reducida o RFD (*Reduced Function Device*). El estándar *ZigBee*, por su parte, utiliza una terminología ligeramente diferente, generando más tipos de dispositivos basados en la descripción del IEEE 802.15.4.

ZigBee define tres tipos de dispositivos: ZC (*ZigBee Coordinator*), ZR (*ZigBee Router*) y ZED (*ZigBee End Device*) [22]. La figura 1.8 describe la relación entre los dispositivos del estándar IEEE 802.15.4 y los de *ZigBee*.

Coordinador *ZigBee*: es el tipo de dispositivo más completo. Existe uno por red, independientemente de la topología utilizada. El mismo es el encargado de iniciar la red y definir la frecuencia del canal de comunicación a utilizar. También escoge el PAN ID (*Personal Area Network Identification*), además de administrar los nodos de la red y

almacenar información sobre estos. Es el responsable de la asociación y disociación de dispositivos.

Router ZigBee: es el encargado de dirigir el encaminamiento de los mensajes entre los nodos de la red. Interconecta dispositivos separados en la topología de la red. Se asocia con el coordinador de la red o con otro *router*.

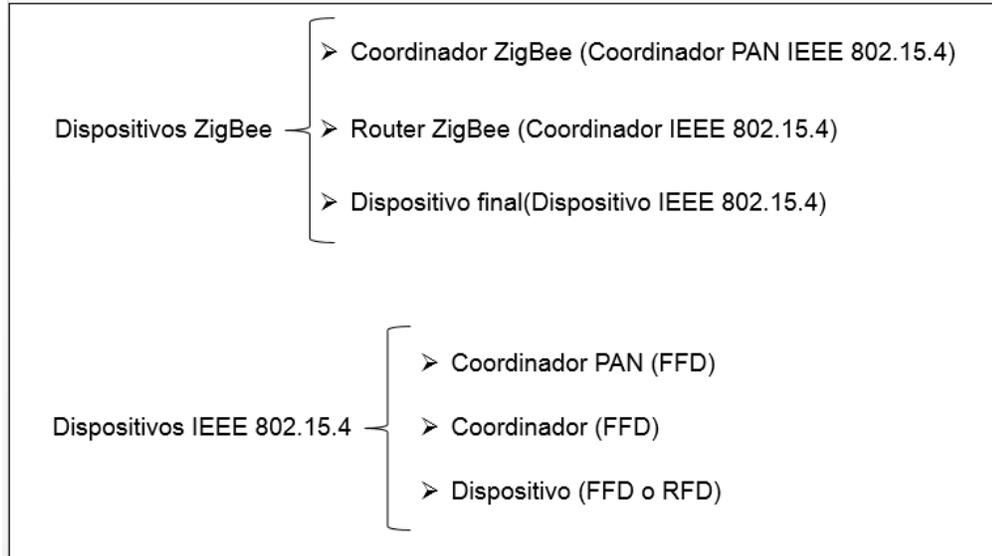


Figura 1.8. Relación entre dispositivos ZigBee y dispositivos IEEE 802.15.4.

Dispositivo final ZigBee: Son dispositivos simples, cuyo papel puede ser asumido por un RFD. Los mismos poseen la funcionalidad necesaria para comunicarse con su nodo padre, pero no pueden transmitir información destinada a otros dispositivos. De esta forma, estos dispositivos pueden estar dormidos la mayor parte del tiempo, aumentando la vida media de sus baterías. Los requerimientos de memoria son mínimos y por lo tanto, son significativamente más baratos. Los dispositivos finales siempre se localizan en los extremos de la red.

Formación de una red ZigBee

Una nueva red *ZigBee* la establece un coordinador. Al inicializarse, el coordinador busca otros coordinadores en sus canales permitidos. Basado en la energía del canal y el número de redes encontradas en sus canales, establece su propia red y selecciona un identificador PAN único de 16 bits. Una vez que la nueva red se establece, los enrutadores y terminales son habilitados a unirse a la red. En caso de conflictos por PAN ID repetidos en diferentes coordinadores, se efectúa un procedimiento de resolución que cambia en uno de ellos su identificador.

Los distintos dispositivos guardan información acerca de otros nodos de la red, en un área no volátil de memoria llamada tabla de vecindades. Al inicializarse, si un dispositivo determina a través de la tabla que fue parte de una red, puede efectuar un procedimiento de notificación para localizarla. Los dispositivos (coordinadores o enrutadores) que reciban notificación, verifican sus tablas para cerciorarse de que el nuevo dispositivo pertenecía a su red. Si la notificación falla o el dispositivo no se encuentra en la tabla de vecindades del resto, trata de unirse a una de las redes como un nuevo dispositivo. Una vez en la red, un dispositivo puede desasociarse ya sea por pedido del coordinador o el *router* (dispositivos padres) o por sí mismo.

Consumo de energía en ZigBee

ZigBee está diseñado con la idea del bajo consumo de energía. Las características que ayudan a reducir el consumo energético son varias. Primero está la tasa de datos: en contraste con Wi-Fi y *Bluetooth*, *ZigBee* no está enviando *e-mails*, documentos grandes, audio, video, entre otros; además, debido a los requerimientos de las aplicaciones de *ZigBee*, los nodos duermen la mayoría del tiempo, ahorrando la energía de la batería.

1.5 Protocolos de enrutamiento en redes de sensores inalámbricos

Para lograr una eficiente comunicación entre los nodos es necesario incluir en la red protocolos de encaminamiento que permitan crear rutas hacia los destinos deseados. Los protocolos tradicionales propios de redes fijas no se adaptan bien a este tipo de entornos tan dinámicos y, por tanto, es necesario el diseño de protocolos específicos para proporcionar un comportamiento eficaz a la red. La principal clasificación de estos protocolos se basa en qué momento se establecen las rutas hacia los destinos deseados. Así, se dividen en protocolos proactivos y reactivos.

Los protocolos proactivos envían periódicamente información de encaminamiento para que en cualquier momento un nodo pueda comunicarse con otros. Esta característica proporciona una rápida respuesta ante solicitudes de ruta y ofrece un buen comportamiento en situaciones donde la tasa de movilidad es alta [2]. Sin embargo, la sobrecarga que se introduce en la red con información de control es alta. Entre estos protocolos se encuentra el DSDV (*Destination-Sequenced Distance Vector*) y DSR (*Dynamic Source Routing*).

Los protocolos reactivos, por su parte, solo crean rutas cuando es necesario. Son protocolos bajo demanda donde la sobrecarga es mucho menor, pero los retrasos de establecimiento de rutas son mayores. Aquí se encuentra por ejemplo AODV (*Ad hoc On-*

Demand Distance Vector). Algunos protocolos, denominados híbridos, mantienen una filosofía proactiva en un ámbito local y reactiva a nivel más global, como el protocolo ZRP (*Zone Routing Protocol*).

1.5.1 DSDV

DSDV es esencialmente una modificación del algoritmo de encaminamiento vector distancia *Bellman-Ford*, bien conocido por su utilidad en redes fijas, como por ejemplo en el protocolo RIP (*Routing Information Protocol*). En este algoritmo, los nodos vecinos intercambian periódicamente sus tablas de encaminamiento enteras para estimar la distancia a la que se encuentran los demás nodos no vecinos.

Las modificaciones introducidas por DSDV proporcionan básicamente la obtención de rutas sin bucles, mediante la introducción de números de secuencia para la determinación de las rutas más nuevas. Aunque DSDV solo proporciona un camino para cada destino, siempre elige el camino más corto basándose en el número de saltos hacia ese destino.

DSDV utiliza dos tipos de mensajes de actualización, uno más grande (*full-dump*) y otro mucho más pequeño (incremental). Los mensajes incrementales se utilizan para actualizaciones intermedias entre envíos periódicos (*full-dump*) de la tabla entera de enrutamiento [2].

1.5.2 DSR

El protocolo DSR se fundamenta en el encaminamiento desde el origen, es decir, los paquetes de datos incluyen una cabecera de información acerca de los nodos exactos que deben atravesar. El mismo no requiere ningún tipo de mensajes periódicos (reactivo), lo que disminuye la sobrecarga con mensajes de control. Además, ofrece la posibilidad de obtener, con la solicitud de una ruta, múltiples caminos hacia el destino [2].

Para realizar el encaminamiento en el origen, a cada paquete de datos se le inserta una cabecera DSR de opciones que se coloca entre la cabecera de transporte y la cabecera IP (*Internet Protocol*). Entre dichas opciones se incluye la ruta que debe seguir el paquete nodo a nodo. Cada nodo mantiene una memoria caché en la que se van almacenando las rutas obtenidas a través de procesos de descubrimiento de rutas, ya sean propias u obtenidas a través de escuchas en la red. En los procesos de descubrimiento se generan mensajes de solicitud, respuesta y error, siendo estos "*route request*", "*reply*" y "*error*" respectivamente.

1.5.3 AODV

En el protocolo AODV los nodos mantienen una tabla de encaminamiento para los destinos conocidos, empleando el algoritmo vector distancia. Inicialmente esta tabla está formada por sus vecinos. Solamente se le añaden destinos nuevos cuando es necesario, es decir, cuando un nodo necesita comunicarse con otro que no está en su tabla, inicia un proceso de descubrimiento de ruta hacia el destino concreto.

Para el descubrimiento de las rutas se emiten mensajes “*rreq*” que se van propagando entre todos los nodos de modo similar al DSR. En cambio, aquí los nodos generan una tabla de encaminamiento inversa para que puedan regresar las contestaciones “*rrep*” a las solicitudes de ruta al nodo que la originó. Este protocolo es capaz además de realizar enrutamiento *multicast*.

1.6 Sistemas operativos para redes de sensores inalámbricos

Las necesidades que tiene un nodo de una red de sensores inalámbricos son totalmente distintas a las que pueda tener cualquier otro dispositivo como puede ser una computadora, por lo tanto, estos motes tienen sus propios sistemas operativos. Los mismos son típicamente menos complejos que los de propósito general, tanto debido a los requisitos especiales de las aplicaciones en las que se usan, como a las restricciones de recursos encontradas en las plataformas *hardware* utilizadas.

1.6.1 TinyOS

TinyOS [24] es un sistema operativo orientado a trabajar con redes de sensores, desarrollado en la Universidad de *Berkeley*. El mismo puede ser visto como un conjunto de programas avanzados, el cual cuenta con un amplio uso por parte de las comunidades de desarrollo, dadas sus características de ser un proyecto de código abierto.

El lenguaje en el que se encuentra programado es un meta-lenguaje que deriva de C, cuyo nombre es *NesC*. Además existen varias herramientas que ayudan al estudio y desarrollo de aplicaciones para las redes de sensores, que van desde aplicaciones para la obtención y manejo de datos, hasta sistemas completos de simulación.

El diseño de TinyOS está basado en responder a las características y necesidades de las redes de sensores, tales como reducido tamaño de memoria, bajo consumo de energía, diversidad en diseños y usos, entre otros. Además se encuentra optimizado en términos de

uso de memoria y eficiencia de energía. El diseño del *kernel* de TinyOS está basado en una estructura de dos niveles de planificación:

- Eventos: Pensados para realizar un proceso pequeño. Además pueden interrumpir las tareas que se están ejecutando.
- Tareas: Pensadas para hacer una cantidad mayor de procesamiento y no son críticas en tiempo. Las tareas se ejecutan en su totalidad, pero la solicitud de iniciar una tarea, y el término de ella son funciones separadas.

El diseño anterior permite que los eventos, que son rápidamente ejecutados, puedan ser realizados inmediatamente, pudiendo interrumpir a las tareas, que tienen mayor complejidad en comparación a los eventos. TinyOS está en continua evolución y sus actualizaciones y novedades se encuentran disponibles para su descarga gratuita en su página web [2].

1.6.2 Contiki

Contiki es un pequeño sistema operativo de código abierto, altamente portable y multitarea, desarrollado para pequeños sistemas, desde ordenadores de 8 bits a sistemas empotrados sobre micro controladores, incluyendo nodos de redes de sensores. A pesar de incluir multitarea y una pila TCP/IP, *Contiki* solo requiere varios kilobytes de código y unos cientos bytes de memoria RAM [2].

El sistema operativo analizado consiste en un núcleo orientado a eventos, el cual hace uso de protohilos, sobre el cual los programas son cargados y descargados dinámicamente. El mismo funciona en una variedad de plataformas, desde micro controladores empotrados, como el MSP430, a viejas computadoras domésticas. El tamaño del código está en el orden de los kilobytes y el uso de la memoria puede configurarse para que sea de solo unas decenas de bytes.

1.6.3 MANTIS

MANTIS (*Multimodal system for Networks of In-situ wireless Sensors*) suministra un nuevo sistema operativo empotrado de plataforma múltiple para redes de sensores inalámbricos. Para conseguir una eficiencia en el uso de la memoria, este sistema es implementado para que utilice una pequeña cantidad de memoria RAM. Este sistema cuenta con un controlador de eficiencia energética que hace que el microcontrolador duerma después de ejecutar todas las tareas activas, lo que reduce el consumo de energía en un rango de micro amperes [25].

La amplia gama de sistemas operativos disponibles para redes de sensores, permite a los administradores escoger el más adecuado, en dependencia de los requerimientos de cada aplicación. Sin embargo, las características propias de las redes de sensores, hacen que la implementación y prueba de nuevas propuestas sea una labor bastante compleja de llevar a cabo en redes de nodos reales, por lo que durante las fases de implementación, depuración y pruebas preliminares de algoritmos y protocolos, lo más adecuado es el uso de un simulador, que facilita en gran medida la detección y corrección de errores de diseño y programación.

1.7 Simuladores para redes de sensores inalámbricos

La simulación es esencial para estudiar las redes de sensores inalámbricas, siendo actualmente uno de los métodos más utilizados para experimentar con nuevas aplicaciones y protocolos. Este hecho favorece la aparición de un gran número de simuladores para este tipo de redes. Los simuladores ayudan en gran medida a los desarrolladores a llevar a cabo la experimentación con nuevos algoritmos, protocolos y aplicaciones antes de ser implantados en el diseño final.

1.7.1 NS2

Network Simulator [26] es un simulador de eventos discretos desarrollado en C++. El mismo es uno de los simuladores de redes no específicos más populares y soporta una gran cantidad de protocolos para diversos tipos de redes. Este simulador se basa en el paradigma de la reusabilidad y proporciona la colección más completa de protocolos de comunicaciones entre los simuladores no comerciales. Incluye un módulo de aplicación llamado *Network Animator* (NAM), que proporciona los resultados de forma visual. Este simulador se explica con más detalle en el capítulo 3.

1.7.2 Omnet ++

Omnet ++ (*Objective Modular Network Testbed in C++*) [27], al igual que NS2, es un simulador de eventos discretos implementado en C++. El mismo proporciona una potente GUI (*Graphical User Interface*), para llevar a cabo el seguimiento y el depurado de las aplicaciones. Comenzar a utilizarlo es muy sencillo, ya que posee un diseño bastante claro, basado en una jerarquía anidada de módulos que proporciona modelos *hardware* incluyendo CPU, batería y radio.

1.7.3 J-Sim

Java Simulator [28] es un simulador basado en componentes, y desarrollado en *Java*. El mismo proporciona simulación de procesos en tiempo real. Su mayor ventaja es la gran cantidad de protocolos soportados, incluyendo un marco de simulación específico para redes de sensores que contiene un modelo muy detallado de estas redes y varias implementaciones para localización, enrutamiento y difusión de datos. Los modelos de J-Sim son fácilmente reutilizables e intercambiables ofreciendo máxima flexibilidad. Además, proporciona una librería GUI para mostrar los resultados de simulación y una interfaz que facilita la creación de *scripts* (archivos de guiones) de simulación.

1.8 Aplicaciones de las redes de sensores inalámbricos

La gran cantidad de parámetros medibles por los sensores, y la posibilidad de recoger los datos de forma desatendida hacen que día a día crezca el número de aplicaciones. El rango de aplicación de estas redes incluye monitorización industrial, automatización de edificios, domótica, medicina, monitorización ambiental, redes de sensores urbanas, gestión de la energía o seguimiento de activos, entre otros.

Generalmente, las aplicaciones se centran en monitorizar parámetros de forma continua en el tiempo, en aquellas zonas en la que no es factible la implementación de una red cableada, ya sea por su alto costo o por conservación del entorno en monumentos o espacios protegidos.

1.8.1 Aplicaciones en la agricultura

La agricultura constituye una de las áreas donde se implementan este tipo de tecnología en los últimos años. Por ejemplo, las redes de sensores favorecen una reducción en el consumo de agua y pesticidas, contribuyendo a la preservación del entorno. Gracias a los desarrollos alcanzados en las redes de sensores inalámbricos en los últimos tiempos, especialmente la miniaturización de los dispositivos, surgen nuevas tendencias en el sector agrícola, como la llamada agricultura de precisión.

Esta disciplina cubre múltiples prácticas relativas a la gestión de cultivos y cosechas, árboles, flores, ganado, entre otras. Por medio de sensores estratégicamente situados, se pueden monitorizar parámetros tales como el clima, la temperatura, la humedad, con el fin de detectar rápidamente situaciones adversas y desencadenar los tratamientos apropiados.

Como un ejemplo de estas aplicaciones se puede mencionar la red que se implanta en los viñedos de *Camalie*, en Estados Unidos, con la cual se tiene uno de los sistemas más avanzados de medida de la humedad del suelo. Utilizan para ello la tecnología inalámbrica desarrollada por la Universidad de Berkeley en colaboración con Intel y comercializada por *Crossbow* [29].

La aplicación consiste en optimizar la irrigación, reduciendo el consumo de agua, la energía utilizada en el bombeado y mejorando la calidad de la uva. Una vez implantado el sistema se comprueba un importante crecimiento en la producción, además de un descenso en el consumo de energía a la hora de utilizar las instalaciones [29].

Varios autores en [30] desarrollan un sistema de control de riego de cultivos en Pakistán. El sistema hace uso de una red inalámbrica de sensores y actuadores para monitorizar parámetros ambientales como temperatura, luz ambiente, humedad del aire y humedad del suelo. Estas medidas son enviadas a través de módulos *ZigBee* a una computadora, que luego de analizarlos, envía las señales de control para activar los actuadores, que en este caso son aspersores.

En [31] se propone el diseño e implementación de un sistema de irrigación de bajo costo basado en tecnología *ZigBee*. El sistema está conformado por un único nodo sensor que mide la temperatura y la humedad del aire y varios nodos actuadores que activan las bombas de agua con comandos de control enviados por un PDA que actúa como un coordinador *ZigBee*.

El manejo adecuado de plagas agrícolas es de vital importancia, de ellos depende que los cultivos no sufran enfermedades que deterioren las plantas. En [32] desarrollan un sistema de adquisición de datos capaz de seguir el comportamiento en espacio y tiempo de las enfermedades del durazno. El sistema hace uso de una red inalámbrica de sensores basada en tecnología *ZigBee* para monitorizar las variables climáticas de temperatura, dirección y velocidad del viento y humedad relativa. Los datos obtenidos son enviados a un dispositivo que los procesa y visualiza para realizar un seguimiento detallado.

Otro ejemplo de la aplicación de este tipo redes en la agricultura lo constituye el proyecto *WiseVine* [5], en el que se implanta una red de 66 nodos en un viñedo, que proporciona interesantes datos para los ingenieros agrónomos encargados del análisis de los resultados.

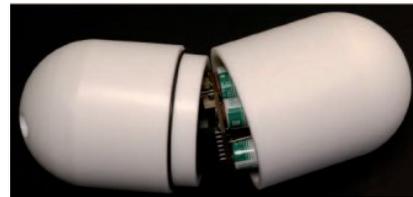
1.8.2 Otras aplicaciones

Prevención de desastres naturales

Con el objetivo de prevenir en la medida de lo posible los desastres naturales, se despliegan diversas redes de sensores en diferentes entornos. Un ejemplo de este tipo de aplicaciones se presenta en [25], donde se propone el uso de una red de sensores para la detección de *tsunamis* y para llevar a cabo el seguimiento de los mismos, con el fin de avisar en primera instancia de su presencia, e informar posteriormente de su recorrido y evolución al tocar tierra.



(a) Estación central para la recogida de datos



(b) Nodos sensores desplegados en el glaciar

Figura 1.9. Monitorización de la evolución de los glaciares (Fuente: [25]).

Otros ejemplos de aplicaciones de este tipo se mencionan en [25], que propone la utilización de una red de sensores para monitorizar la evolución de los glaciares (ver figura 1.9), o [33] que detalla el uso de redes de sensores para la monitorización de barreras de entornos volcánicos (ver figura 1.10). En [25] se detalla también el uso de redes de sensores para la monitorización de barreras de coral.

Control de procesos industriales

Con el objetivo de reducir costes y mejorar el rendimiento de la producción, los sensores son utilizados en el marco de la industria para supervisar el estado de la maquinaria, controlando el nivel de desgaste, lubricación o vibración. Otro de los usos en la industria son los controles de calidad monitorizados por redes de sensores. La Universidad de Berkeley desarrolla un experimento en el que los sensores son situados a lo largo de una

cadena de montaje en una factoría de automóviles para vigilar la operación de los robots, facilitando su mantenimiento y evitando problemas derivados de las redes cableadas [25].

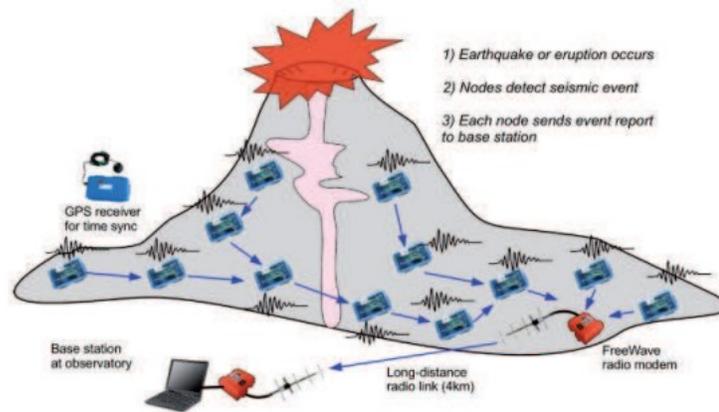


Figura 1.10. Redes de sensores para la monitorización de entornos volcánicos (Fuente: [34]).

Medicina

Los sensores inalámbricos también pueden ser utilizados para monitorizar las constantes vitales de los pacientes, ya sea dentro del propio hospital, durante el traslado hacia el centro sanitario o en su propio domicilio en períodos de convalecencia. El proyecto *Vital Sign Monitoring* [34] crea un sistema que consta de cuatro componentes: un identificador de pacientes, compuesto por un nodo especial que contiene los datos del paciente, sensores médicos, que miden parámetros como las pulsaciones por minuto o la presión arterial, un dispositivo de visualización de datos y otro de programación del sensor.

Los datos recogidos se envían periódicamente a una estación base que los almacena en una base de datos. Posteriormente, los facultativos se encargan de realizar los diagnósticos y tratamientos oportunos en función de la información obtenida de cada paciente.

CAPITULO 2 . PLATAFORMAS DE REDES DE SENSORES INALÁMBRICOS

En el mercado existe una gran cantidad de plataformas comerciales disponibles para redes de sensores inalámbricos. En algunos casos se diseñan de carácter específico, destinadas a realizar una función concreta y que, por tanto, no se pueden adaptar a otros entornos. Por otro lado, se diseñan nodos de propósito general, los cuales pueden ejecutar diferentes medidas y utilizar en múltiples aplicaciones.

En este capítulo se presenta una revisión a algunas de las principales plataformas desarrolladas para aplicaciones de redes de sensores inalámbricos, así como las puertas de enlace y placas de sensores disponibles para las mismas.

2.1 Plataforma MICAz

Los motes MICAz [35], desarrollados por la compañía Crossbow [36], son dispositivos para el desarrollo de aplicaciones de medición, control de acceso y demás aplicaciones de redes de sensores inalámbricos. Para la transmisión de datos, los mismos cuentan con el chip de radio CC24240, el cual ofrece una tasa de transmisión de hasta 250 Kbps y opera en la banda ISM de los 2.4 GHz.



Figura 2.1. Plataforma MICAz (Fuente: [35]).

La plataforma, mostrada en la figura 2.1, tiene una sensibilidad de -94 dBm, y el alcance de radio, con una antena de media onda, está entre 75 y 100 metros en exteriores con línea

de vista, y entre 20 y 30 metros en interiores. La misma soporta funciones de enrutamiento, ofreciendo la posibilidad de crear redes con topología estrella y *mesh*.

Cada mote MICAz incorpora un microprocesador Atmel ATmega128L de 7 MHz. Dicho procesador tiene una memoria EPROM (*Erasable Programmable Read-Only Memory*) de 128 Kbytes de espacio para los programas y otra EPROM de 4 Kbytes para datos.

La compañía desarrolla dos tipos de placas, unas con sensores integrados y otras a las que pueden conectarse sensores externos a través de los conversores analógicos digitales. En el caso de esta última opción resulta difícil en ocasiones encontrar sensores que se adapten a las características de este tipo de placas, debido a que el voltaje de salida debe estar en el rango de 0 a 3 V. En la tabla 2.1 se listan las placas sensoras disponibles para esta plataforma.

Tabla 2.1. Placas sensoras para los nodos MICAz (Fuente: [37]).

Placa sensora	Sensores
MTS310CB	Luz, temperatura, micrófono, zumbador, acelerómetro y magnetómetro
MTS400CB	Luz ambiente, humedad relativa, temperatura, acelerómetro, y presión barométrica
MTS420CC	Lo mismo que el MTS400CB pero con un módulo GPS
MDA100CB	Luz, temperatura y un área para prototipos
MDA300CA	Luz, humedad relativa e interface de propósito general para sensores externos

Un aspecto importante en este tipo de redes es el consumo de energía de los nodos sensores, puesto que los mismos no están conectados a la red eléctrica y su funcionamiento se basa en la utilización de baterías. El consumo energético de los motes es de 19.7 mA en recepción y de 17.4 mA en transmisión.



Figura 2.2. Placa programadora MIB520CB (Fuente: [35]).

Estos nodos están diseñados para trabajar con una tensión de 2.7 V, por lo que se pueden alimentar con un par de baterías tipo AA. Para lograr mayores períodos de funcionamiento, puede tomarse la opción de utilizar la energía solar para alimentar los motes.

Para la programación de esta plataforma se utiliza la placa MIB520CB, mostrada en la figura 2.2. Esta placa tiene dos funcionalidades básicas. Por un lado permite cargar el código ejecutable en los nodos, y por otro puede funcionar como puerta de enlace entre los nodos y una computadora.

2.2 Plataforma Mica2

Los nodos Mica2 [38], mostrados en la figura 2.3, son los módulos de tercera generación de la familia Mica que se usan para redes de sensores inalámbricos de baja potencia. Entre sus principales aplicaciones se encuentran seguridad y vigilancia, monitorización ambiental, entre otras. Ellos mejoran las características del Mica original en los siguientes aspectos:

- Diseñado específicamente para redes de sensores integradas.
- Distintas frecuencias de transmisión con amplio rango.
- El sistema operativo es distribuido por TinyOS, mediante un software v1.0 y una plataforma mejorada para las redes de sensores que contiene mejores características que eliminan errores.
- Permiten reprogramación inalámbrica a distancia.
- Amplia gama de tarjetas con sensores y adquisición de datos adicionales.
- Compatible con los motes Mica2Dot.

Esta plataforma cuenta con un microcontrolador Atmega128L, que descarga y ejecuta las aplicaciones en su memoria interna con ayuda de la puerta de enlace MIB510. El mismo procesa a una velocidad de 4MHz y consume únicamente, al igual que el MICAz, 8 mA en estado de procesamiento y solo 15 μ A en modo *sleep*. Este bajo consumo de energía permite que el mote trabaje por períodos mayores a un año con un par de baterías tipo AA.

Este mote está provisto de una memoria *flash* de 512 Kbytes para almacenar los datos adquiridos. También cuenta con un conversor analógico-digital (A/D) de 10 bits para que los datos del sensor puedan ser digitalizados. Las tarjetas de sensores MTS pueden adaptarse a la plataforma mediante el conector de 51 pines.

Para la transmisión y recepción de los datos utiliza el chip de radio CC1000, el cual opera en la banda ISM y en la banda SRD (*Short Range Device*). Este chip permite una velocidad de transmisión aproximada de 40 Kbps. Cabe mencionar que debido a que el radio tiene un receptor muy sensible, los motes deben colocarse a una distancia mínima de 60 cm.

En dependencia de la frecuencia de trabajo, los Mica2 se dividen en tres modelos: el MPR400, que opera en la banda de los 915 MHz, MPR410 en los 433 MHz y el MPR420 en la frecuencia de los 315 MHz.



Figura 2.3. Plataforma Mica2 (Fuente: [38]).

Varios sensores y tarjetas de adquisición de datos están disponibles para esta plataforma. Entre estas placas se encuentra la MTS300CA/MTS310CA, mencionada para el caso de los nodos MICAz y la placa MTS101CA, la cual cuenta con sensores de temperatura, luz y un área para prototipos. Para la programación de estos motes se emplea, al igual que con las plataformas MICAz, la placa de desarrollo MIB520CB y además la placa MIB510, la que sirve además de puerta de enlace.

2.3 Plataforma Mica2Dot

Los Mica2Dot [39] son un tipo de motes diseñados especialmente para aplicaciones donde el tamaño físico es fundamental, ya que tienen un diámetro aproximado de 3 cm. Al igual que en los motes Mica2, cuentan con un microcontrolador Atmega128L, que permite la descarga y ejecución de aplicaciones, y un transceptor CC1000 que realiza la transmisión y recepción de datos.

Al igual que en los Mica2, existen tres modelos en dependencia de la frecuencia de trabajo: MPR500, a la frecuencia de 915 MHz, MPR510 en los 433 MHz y el MPR520, que opera en los 315 MHz. El resto de las características son similares a las de los Mica2, lo más relevante es la forma física y el reducido tamaño que poseen, tal y como se puede apreciar en la figura 2.4.

Al contrario de los motes Mica2 y MICAz, no posee el conector de expansión de 51 pines, en su lugar tiene un conector de 18 pines para conectar 6 entradas analógicas, E/S (Entrada/Salida) digitales y una interfaz de comunicaciones serie.

En cuanto a las placas sensoras compatibles con este mote se encuentra el MTS510CA y MDA500CA. En el caso del MTS510CA, es una tarjeta de sensores flexible con una variedad de modalidades de sensado. La misma contiene sensores de luz, micrófono y acelerómetro. En cuanto al MDA500CA, provee un área para prototipos, lo que permite la conexión de señales externas al Mica2Dot.

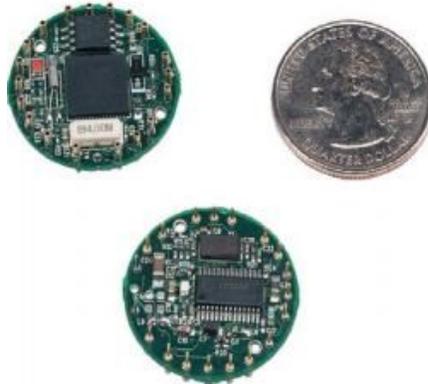


Figura 2.4. Motes Mica2Dot (Fuente: [39]).

2.4 Plataforma Imote2

Intel Mote [40] , ver figura 2.5, surge de una colaboración entre los laboratorios de Intel en Berkeley, la Universidad de Berkeley y otros centros de investigación. Los motes de Intel son pequeños, autónomos, alimentados por medio de baterías y con comunicación vía radio, de forma que son capaces de compartir información con otros y organizarse automáticamente dentro de una red *ad-hoc*.

Uno de los principales objetivos del grupo de trabajo del Intel Mote es colaborar con la comunidad investigadora en la exploración de las potenciales aplicaciones de los nodos y de las redes de sensores. Con este objetivo en mente, se diseñan los nodos con una total compatibilidad con el sistema operativo TinyOS.

La plataforma contiene dos interfaces de sensores básicos en uno de los lados de la placa central, y otras dos interfaces avanzadas para sensores en el otro sector. La placa principal contiene el procesador y el módulo de radio, y se le pueden agregar diferentes módulos sensores dependiendo de la aplicación final de la red.

El procesador que contiene la placa principal puede funcionar a bajo voltaje (0.85V) y a una frecuencia baja de 13MHz cuando se habilita el modo de bajo consumo, mientras que el valor máximo de operación es de 416MHz. También integra una memoria SRAM (*Static Random Access Memory*) de 256KB, dividida en 4 bancos de 64KB y diferentes opciones

de E/S, lo que lo hace extremadamente flexible para soportar diferentes sensores analógicos/digitales.



Figura 2.5. Plataforma Imote2 (Fuente: [40]).

Las E/S mencionadas incluyen I2C (*Inter-Integrated Circuit*), 3 puertos serie síncronos, 3 puertos UART (*Universal Asynchronous Receiver-Transmitter*), E/S digitales y un cliente USB (*Universal Serial Bus*). Además, el propio procesador incluye diversos temporizados y un reloj en tiempo real. El principal problema de este dispositivo es su elevado precio, el cual está alrededor de los mil euros.

2.5 Plataforma Iris

Los nodos Iris [41] pertenecen a la compañía *Crossbow Technology*. Esta plataforma utiliza el microcontrolador ATmega1281, y trabaja en la banda de frecuencias de 2405 MHz a 2480 MHz. Los mismos permiten una velocidad de transmisión de datos de 250 Kbps.



Figura 2.6. Plataforma Iris (Fuente: [41]).

Esta tecnología brinda hasta tres veces más de alcance de radio, y el doble de memoria de programa en comparación con la generación anterior de los motes Mica. Al igual que la familia de motes Mica, cuenta con un conector de expansión de 51 pines, y utilizan la misma memoria *flash* de serie. En la figura 2.6 se muestra esta plataforma.

2.6 Plataforma Tmote Sky

Tmote Sky [42] es una plataforma también comercializada por *Crossbow*, la cual se usa para aplicaciones en redes de sensores de muy bajo consumo y alta recopilación de datos. Ella lleva integrados tanto los sensores como la radio, antena y microcontrolador, además puede ser fácilmente programada. En la figura 2.7 se puede observar la plataforma y cada una de sus partes componentes.

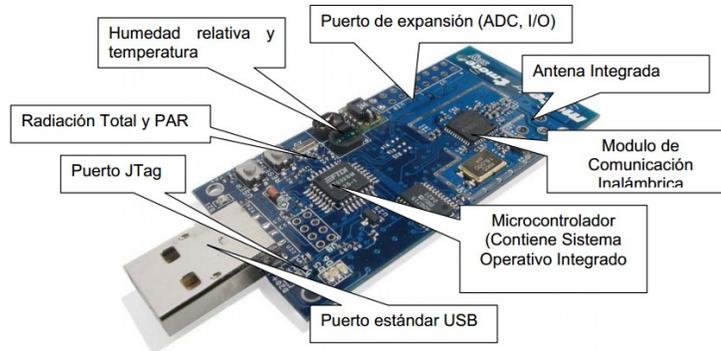


Figura 2.7. Plataforma Tmote Sky (Fuente: [42]).

Las operaciones de baja energía son realizadas gracias al microcontrolador MSP430 F1611. Este procesador RISC (*Reduced Instruction Set Computer*) de 16 bits consume muy poca batería tanto en el estado activo como durante el sueño o hibernación.

Para la transmisión, este mote utiliza el transceptor CC2420, que cumple con el estándar IEEE 802.15.4 y que permite una tasa de transmisión de hasta 250 Kbps, con un alcance de radio que puede llegar a los 125 metros.

A nivel de memoria, la plataforma analizada es sensiblemente superior a los nodos de la familia Mica, ya que incorpora un microcontrolador con una memoria RAM de 10 Kbytes y un módulo de memoria *flash* externo de 1Mbyte.

Tmote Sky no incorpora ningún conector de expansión para sensores, solamente la posibilidad de incorporar un sensor de humedad y uno de temperatura. El mismo puede ser alimentado con dos baterías tipo AA.

2.7 Plataforma eKo

En 2008 la empresa *Crossbow* lanza al mercado el *kit eKo* [43], el cual presenta importantes mejoras en cuanto a robustez con respecto a productos anteriores y que es, en la actualidad, su producto estrella en el ámbito de la agricultura. El fabricante lo define como

un sistema inalámbrico de monitorización ambiental de exterior para aplicaciones de agricultura de precisión, microclima y estudios de conservación, investigación medioambiental y monitorización de cultivos.

La plataforma estudiada tiene una sensibilidad de -101dBm , y un consumo de 16 mA en recepción y 17 mA en transmisión, lo cual permite extender la duración de las baterías por alrededor de dos años sin necesidad de recargarlas.



Figura 2.8. Vista general del kit eKo (Fuente: [43]).

La figura 2.8 muestra 4 nodos del kit, la estación base del sistema, que conectada con el eKo gateway (esquina inferior derecha) permite el almacenamiento local de los datos monitorizados e interactuar con el sistema a través de Internet.



Figura 2.9. Sensores comercializados para el kit eKo (Fuente: [43]).

Los nodos sensores y la estación base se comercializan en dos versiones que operan en la banda de frecuencia de los $2,4\text{ GHz}$. La versión de corto alcance ofrece coberturas de entre 152 y 457 m en línea de vista. La de largo alcance tiene coberturas de entre 610 y 3218 m .

Los nodos sensores incluyen un pequeño panel solar que recarga 3 baterías de tipo AA. Además, los nodos disponen de 4 conectores externos en su base que permiten conectar hasta 4 sensores comercializados por el fabricante. La conexión de los sensores se realiza por el denominado *bus plug-and play* ESB (*Environmental Sensor Bus*) desarrollado por el fabricante.



Figura 2.10. Vista de la estación meteorológica conectada a un nodo del kit eKo (Fuente: [43]).

La figura 2.9 muestra 3 de los 5 sensores que comercializa el fabricante para el *kit eKo*. De izquierda a derecha se tratan de un sensor de temperatura y humedad del suelo, una sonda de humedad relativa y temperatura del ambiente y un sensor de radiación solar.

Además de los sensores anteriores, al sistema también se puede conectar una estación meteorológica (ver figura 2.10) que comercializa el fabricante. La estación incluye sensores de pluviometría, temperatura y humedad del ambiente, radiación solar, presión barométrica, velocidad y dirección del viento, y un anemómetro.

2.8 Plataforma *Waspote*

La empresa *Libelium* [44], comercializa desde el año 2009 un nodo sensor para agricultura de precisión. En concreto, el dispositivo permite controlar hasta catorce parámetros ambientales diferentes en viñedos y casas de cultivo. El elemento principal del sistema es la plataforma *Waspote* [45] (ver figura 2.11), la cual permite integrar varios sensores diferentes para monitorizar diversos tipos de entorno.

Los datos adquiridos por el dispositivo analizado pueden ser enviados inalámbricamente a doce kilómetros de distancia usando el protocolo *ZigBee* o almacenados en el propio

dispositivo. Este alcance se debe a que la potencia de transmisión se encuentra entre los 50 y los 100 mW.

La empresa distribuye diferentes placas de sensores que pueden ser conectadas al dispositivo. En la figura 2.12 se muestra la placa de sensores para agricultura de precisión. Con esta placa se pueden monitorizar hasta catorce parámetros medioambientales: temperatura y humedad relativa del aire, temperatura de suelo, humedad de suelo, humectación de hoja, presión atmosférica, radiación solar, diámetro de tronco/tallo/fruto, velocidad de viento, dirección de viento y precipitación.



Figura 2.11. Plataforma Waspote (Fuente: [45]).

En lo que respecta a la conectividad vía radio, se le pueden añadir diferentes módulos XBee o bien un módulo *Bluetooth*. También permite añadir un módulo GSM (*Global System for Mobile communications*)/ GPRS (*General Packet Radio Service*) para el envío y recepción de datos a través de la red GSM. Además es posible añadir un módulo GPS (*Global Position System*) con su correspondiente antena.



Figura 2.12. Placa de sensores para agricultura de precisión (Fuente: [45]).

Junto con la plataforma *Waspote* se comercializa además un dispositivo *gateway*, denominado *Meshlium*, que permite comunicarse con los nodos vía radio, *bluetooth*, o GSM/GPRS y hacer de pasarela hacia internet. Además de esta puerta de enlace también

se cuenta con un el *Waspote Gateway*, que permite la conexión entre las plataformas y una computadora.

Además de las facilidades mencionadas que ofrece esta plataforma, la empresa comercializa un dispositivo denominado *Waspote Plug & Sense*, el cual ofrece la posibilidad a los desarrolladores de olvidarse de la parte electrónica y concentrarse en los servicios y aplicaciones deseadas. La nueva plataforma consiste de una robusta carcasa a prueba de agua con *sockets* específicos para la conexión de sensores, panel solar, la antena y el cable USB.

Las primeras plataformas analizadas (MICAz, Mica2, Mica2Dot, Iris) son productos comerciales que incluyen sensores de bajo costo y cuyas especificaciones (precisión, resolución, entre otros), se alejan de las de la instrumentación que habitualmente se utiliza en agricultura de precisión. Por otro lado, muchos de estos nodos se utilizan en aplicaciones de laboratorio o proyectos de investigación, por lo que carecen de la robustez necesaria para su despliegue en entornos agrícolas.

Además de las consideraciones anteriores, los nodos MICA2 y Mica2Dot no pueden ser utilizados para la realización del presente proyecto, pues trabajan en las bandas de frecuencia de los 868 MHz y 915 MHz. Esta porción del espectro está reservada para el uso en Europa (868 MHz) y los Estados Unidos (915 MHz).

Por otra parte, la plataforma Imote2, está destinada para aplicaciones industriales, donde los requerimientos de almacenamiento y procesamiento son mucho mayores que los necesarios en aplicaciones agrícolas.

Las plataformas *eKo* y *Waspote* amplían las posibilidades para monitorizar parámetros medioambientales, además de presentar mejores características en cuanto a robustez frente a las inclemencias del medio donde se despliegan. En estos momentos son los productos estrellas para redes sensoriales aplicadas a entornos agrarios.

La placa sensorial *Waspote* permite la conexión de más de 10 sensores al mismo tiempo, sin embargo los requerimientos de energía son mayores, pues están diseñados para lograr alcances de hasta 12 km. Una red sensorial con esta plataforma puede medir la efectividad de la fertirrigación, el crecimiento de las cosechas y las condiciones micro-climáticas, además de detectar sucesos meteorológicos adversos.

El objetivo del presente proyecto de tesis es diseñar una red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”, la cual tiene un área total de 1,8 hectáreas. Las distancias entre los nodos más alejados y la estación base se encuentran alrededor de los 100 metros, por lo tanto, no se necesita que el alcance de radio vaya más allá de estas distancias. Las variables que se desean monitorizar son la humedad relativa, la radiación solar y la temperatura. Bajo estas premisas se decide escoger la plataforma *eKo* para la realización del diseño.

CAPITULO 3 . DISEÑO DE LA DE RED DE SENSORES INALÁMBRICOS PARA LA “CASA DE CULTIVOS PROTEGIDOS SAN JOSÉ”

La agricultura juega un papel crucial en el desarrollo de la civilización humana. El incremento de la demanda de alimentos a nivel mundial en los últimos años, provoca una tendencia a aplicar esfuerzos extras y técnicas especiales para multiplicar la producción de nutrientes. El desarrollo de áreas de cultivos protegidos es un ejemplo de estos empeños.

En este capítulo se presenta la descripción de la “Casa de Cultivos Protegidos San José”, así como el diseño propuesto de la red de sensores inalámbricos para esta instalación y la simulación del mismo en el NS2.

3.1 “Casa de Cultivos Protegidos San José”

La “Casa de Cultivos Protegidos San José” se encuentra ubicada en el municipio Santiago de Cuba, perteneciente a la provincia del mismo nombre. Posee un área total de 1,8 hectáreas y está conformado por 23 casas de cultivo y una casa de posturas. En la figura 3.1 se muestra una vista de algunas de las casas de cultivo, mientras que en la figura 3.2 se observa la casa de posturas.



Figura 3.1. Vista de las casas de cultivos protegidos de San José.

Las casas de cultivos poseen 45 metros de largo y 10 metros de ancho, para un área de 450 m², mientras que la casa de posturas tiene un ancho de 10 metros y un largo de 15 metros, para un área de 150 m². El recubrimiento es de *nylon* y las mallas laterales son antiafidas. Estas estructuras presentan además apertura cenital [13].



Figura 3.2. Casa de posturas.

La instalación está formada por dos hileras de casas de cultivos, separadas una distancia de 40 metros; la primera hilera está compuesta por doce naves, mientras que la segunda cuenta con once naves y la casa de posturas. Las casas de cultivos están separadas una distancia de 4 metros, excepto entre las casas número 4 y 5 y la 16 y 17, pues en este caso la separación es de 10 metros. En la figura 3.3 se puede observar una vista general del establecimiento.

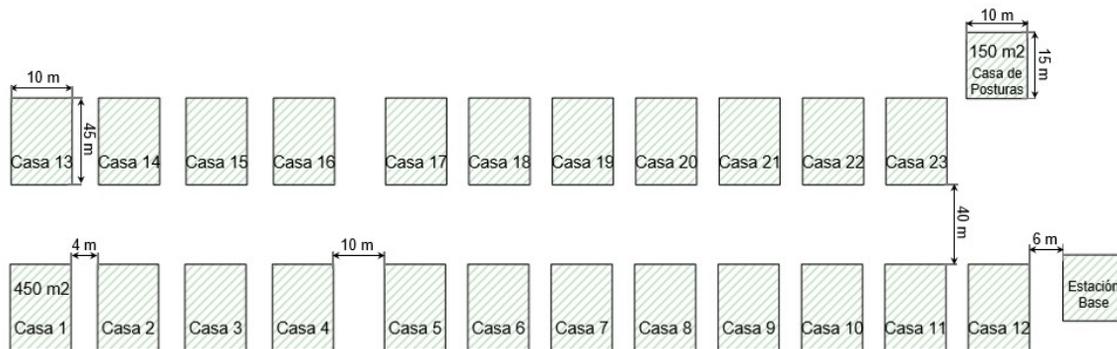


Figura 3.3. Vista general de la infraestructura.

Las principales especies cultivadas son el pepino, tomate, pimiento, sandía, melón, col, lechuga, entre otros. El destino principal de estos productos es el turismo, con lo cual se

sustituyen costosas importaciones. En las figuras 3.4 y 3.5 se observa una plantación de lechuga y tomate respectivamente.



Figura 3.4. Plantación de lechuga.

3.2 Control climático en casas de cultivo

El control climático de las casas de cultivo implica que en las mismas se mantengan las condiciones necesarias para que la flora allí existente pueda desarrollarse de la mejor manera y realizar una producción abundante y de buena calidad. Los cultivos en sus diferentes etapas de desarrollo están condicionados principalmente por tres factores: luz, humedad relativa y temperatura. Las casas deben conservar estas variables dentro de rangos para mantener el metabolismo de los vegetales.



Figura 3.5. Plantación de tomate.

3.2.1 Temperatura

La temperatura es el parámetro más importante a tener en cuenta en el manejo o control del ambiente dentro de una casa de cultivo. Afecta directamente las funciones vitales como fotosíntesis, respiración, permeabilidad de la membrana celular, absorción de agua y nutrientes, transpiración, actividades enzimáticas, entre otros, procesos metabólicos que intervienen en el crecimiento y desarrollo de las plantas [46].

Tabla 3.1. Exigencias de temperatura para distintas especies (Fuente: [46]).

	T ^o mínima. letal	T ^o máxima. letal	T ^o óptima
Sandía	0	33-37	17-20
Melón	0-1	33-37	18-21
Pepino	-1	31-35	18
Pimiento	1	33-35	16-18
Tomate	0-2	33-38	13-16

Donde:

- Temperatura mínima/máxima letal: Temperatura por debajo/arriba de la cual se producen daños en la planta.
- Temperatura óptima: Indican los valores aconsejados para un correcto funcionamiento de la planta.

Las reacciones biológicas de importancia no pueden desarrollarse si la temperatura está por debajo de 0°C, o por encima de 50°C. El límite inferior corresponde al punto de congelación del agua y el superior a la desnaturalización de las proteínas. La temperatura óptima varía según las especies, pero casi siempre está comprendida entre 10° y 25°C. Sin embargo, para realizar un mejor control de esta variable es sumamente importante conocer y regirse por las necesidades y limitaciones de la especie que se está cultivando. En la tabla 3.1 se enumeran las necesidades y limitaciones de algunas de las especies que se cosechan en las casas de cultivo de San José.

3.2.2 Humedad relativa

La humedad es la masa de agua en una unidad de masa de aire. La humedad relativa se refiere a la cantidad de agua contenida en el aire, en relación con la que se podría contener a la misma temperatura. La relación existente entre temperatura y humedad es de tipo inversa, por lo tanto, a mayor temperatura, la capacidad de almacenar agua será menor [47].

La humedad relativa dentro de las casas de cultivo interviene en varios procesos, como el amortiguamiento de los cambios de temperatura, el crecimiento de las plantas, la viabilidad del polen para obtener mayor porcentaje de fecundación y el desarrollo de enfermedades. Dado que las especies de vegetación que se puede cultivar son diferentes en sus requerimientos de temperatura, también difieren en sus necesidades de humedad relativa. En la tabla 3.2 se muestran los requerimientos de algunos cultivos.

Tabla 3.2. Requerimientos de humedad relativa (Fuente: [47]).

	Sandía	Melón	Pepino	Pimiento	Tomate
Humedad	65-80%	60-70%	70-90%	50-60%	50-60%

3.2.3 Iluminación

La iluminación es otro de los factores climáticos que también puede influir de manera considerable en el resultado de los cultivos, ya que incide directamente sobre el efecto de la fotosíntesis. Para una fotosíntesis máxima es necesario que cuando se tenga mayor luminosidad se aumenten los niveles de temperatura y humedad relativa. Cuando existe poca iluminación los requerimientos de los otros parámetros también disminuyen.

Un control climático productivo de las casas de cultivo implica manejar correctamente todos los sistemas que se disponen, para realizar el control de las variables mencionadas, con lo cual se podrá mantener un desempeño óptimo de la producción. Existen otras áreas del funcionamiento de las casas de cultivo, entre ellas los sistemas de riego, que también requieren atención y que describirían otros requerimientos adicionales de esta técnica de cultivo.

3.3 Diseño de la red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”

Las redes de sensores inalámbricos son altamente dependientes de la aplicación, por lo que es necesario considerar cuidadosamente los requisitos específicos del diseño que se desea implementar, especialmente cuando se es tan exigente como el monitoreo ambiental.

Tales aplicaciones generalmente consisten en la implementación de una serie de sensores en campo para medir periódicamente los parámetros meteorológicos, como la temperatura y la humedad. En esta sección se explican las especificaciones y los requerimientos del sistema.

3.3.1 Especificaciones de diseño

Una vez analizadas las características del campo de desarrollo y de la plataforma a utilizar, se tienen las siguientes especificaciones del sistema:

- Los nodos sensores no tendrán movilidad.
- La latencia en la comunicación no es un factor restrictivo, ya que el sensado de variables medioambientales no presenta limitaciones críticas. Sin embargo, se deben mantener valores adecuados de latencia, para que en un futuro avance de esta investigación, se puedan introducir mecanismos que controlen las variables de desempeño de las casas de cultivo.
- En el interior de las casas de cultivo, la interferencia por obstáculos es aceptable, ya que la interferencia de las plantas no es significativa.
- En campos cerrados, la diferencia en los factores climáticos de un sector a otro tienden a nivelarse rápidamente, haciendo menores los requerimientos de sensado.
- Los factores meteorológicos que se desean monitorizar cambian lentamente en el tiempo, lo que permite el muestreo escaso (típicamente cada diez a quince minutos es suficiente).
- El alcance de radio de los nodos, no permite en este caso utilizar la topología en estrella. Por lo tanto, existe la necesidad de emplear la topología malla o *cluster-tree*.

Tomando en cuenta estas consideraciones se propone la utilización de 24 nodos sensores, ubicados en cada una de las casas de cultivo y en la casa de posturas. La estación base consta de una pasarela, la cual permite transmitir los datos a una computadora, y un nodo que actúa como coordinador de la red PAN, el cual es el encargado de recolectar la información proveniente de los nodos finales. La ubicación física de cada uno de los motes se muestra en la figura 3.6. A la izquierda se observa la distribución en las casas de la número 1 a la 22, en el centro la de la casa 23 y a la derecha la ubicación en la casa de posturas. La diferencia entre la nave 23 y el resto se debe al alcance de radio entre esta y la estación recolectora.

Un aspecto importante a tener en cuenta es el ahorro energético, de ahí que es indispensable evaluar cuál de las topologías presenta las mejores características. Para realizar esta evaluación se emplea el simulador NS2, el cual permite la simulación de redes de sensores inalámbricos.

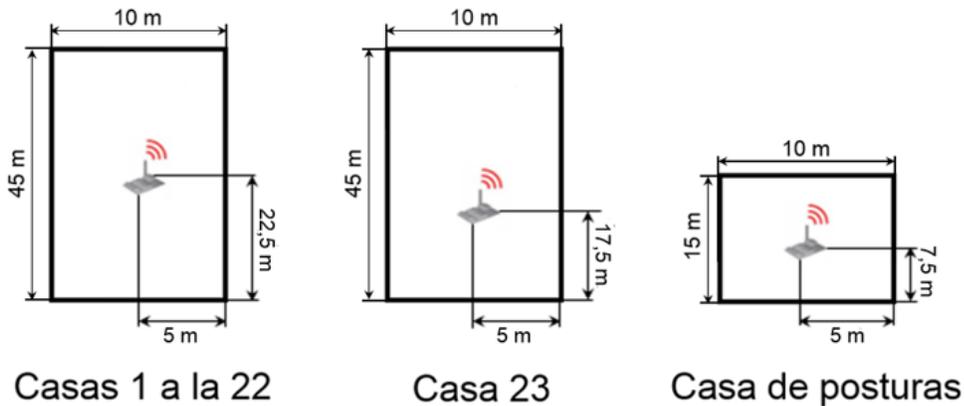


Figura 3.6. Ubicación de los nodos sensores.

3.4 Simulación de las topologías de red

En este epígrafe se analiza la viabilidad de cada una de las topologías de red permitidas por el estándar *ZigBee*, las cuales son: estrella, malla y basada en *cluster*. La utilización de la topología en estrella no es posible, debido a la distancia que existe entre los nodos más alejados y la estación base, lo cual impide que los nodos puedan sincronizarse con el coordinador de la red PAN. La distribución física de los nodos es la misma para los dos casos analizados. A continuación se detallan las principales características del simulador empleado y las herramientas complementarias.

3.4.1 *Network Simulator 2 (NS2)*

NS2 es un software para la simulación de eventos discretos, ideado para la ayuda a la investigación de redes. NS proporciona soporte para la simulación de multitud de protocolos de las capas de aplicación, transporte y protocolos de enrutamiento, tanto para redes cableadas como inalámbricas o vía satélite, además de simular topologías complejas con un gran número de generadores de tráfico [26].

NS es básicamente un simulador orientado a objetos, escrito en C++, cuya interfaz de usuario se presenta como un intérprete de lenguaje tcl orientado a objetos o, en otras palabras, de lenguaje Otcl. El simulador soporta una jerarquía de clases escrita en C++, también llamada jerarquía compilada, y otra jerarquía de clases similar a la anterior, pero dentro del intérprete, que se presenta al programador en Otcl y que se conoce como jerarquía interpretada. Estas dos jerarquías están estrechamente relacionadas entre sí, de modo que, desde la perspectiva del usuario, hay una correspondencia uno a uno entre una clase de la jerarquía interpretada y otra de la compilada [26].

El simulador analizado cuenta con herramientas complementarias como el NAM (*Network Animator*) y XGRAPH. La utilidad de NAM es que puede representar gráficamente la red, que previamente se construye mediante comandos escritos en un lenguaje de programación llamado tcl y posteriormente compilados por NS. Así mismo, NAM puede visualizar dinámicamente el desplazamiento de los paquetes de la simulación y que NS almacena en un archivo junto con la propia topología de la red. Estos resultados que NS produce a partir de los archivos que contienen los comandos escritos en lenguaje tcl, mediante *scripts* (archivos de guiones), lógicamente, dependen de la topología, protocolos, parámetros, y demás actividades que en ellos se definan [26].

Por otra parte, con las gráficas bidimensionales de XGRAPH se pueden hacer análisis de paquetes recibidos en la comunicación, paquetes perdidos, ancho de banda y de retardos. Para el estudio de los resultados de las simulaciones que se realizan en el presente proyecto de tesis, no es posible la utilización de esta herramienta, debido a que las trazas generadas en las simulaciones de redes de sensores inalámbricos no son compatibles con la versión del programa disponible en estos momentos.

3.4.2 Estructura de simulación

Para la simulación de las topologías de red se crean los escenarios, los cuales permiten obtener las variables para poder analizarlos y comparar las topologías. Para lograr este objetivo, el usuario define, mediante un *script*, la configuración del simulador, las características de la red inalámbrica, y otros aspectos básicos del tipo de red a simular, como son las características de los nodos, el tipo de tráfico a utilizar, entre otros. En la figura 3.7 y 3.8 se muestran la estructura básica de simulación y los parámetros principales que deben ser configurados en los *scripts* respectivamente.

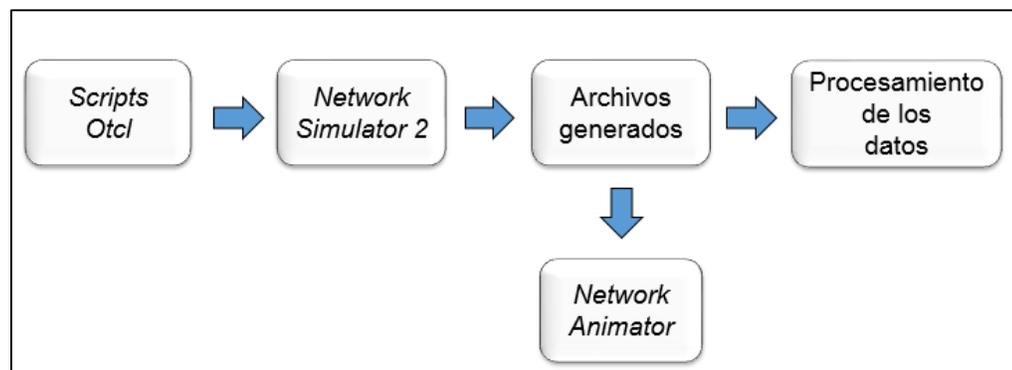


Figura 3.7. Estructura básica de simulación.

Una vez que se crea el *script*, se ejecuta un intérprete de comandos. NS2 abre y ejecuta el archivo que se pasa como argumento (ver figura 3.9), el mismo debe tener extensión tcl y en él deben estar todas las instrucciones necesarias para realizar la simulación.

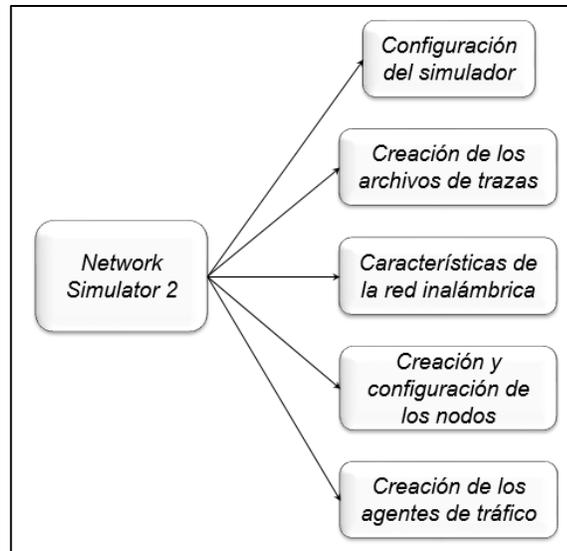


Figura 3.8. Creación del script de simulación.

Los resultados obtenidos de las simulaciones se almacenan en archivos de trazas (.tr y .nam). La evaluación de las topologías de red depende de la interpretación de estos ficheros, por lo que resulta importante comprenderlos. Estos archivos son difíciles de entender, debido a su gran tamaño y multitud de datos sin explicación alguna, es por ello que se precisa de otros programas para razonar e interpretar los resultados obtenidos. Algunos son de representación gráfica, como el NAM, explicado anteriormente, otros son programas de filtrado como awk¹, el cual se utiliza en este proyecto para mostrar solo la información deseada.

```
felix@felix-Aspire-5720Z:~/Documentos/Simulaciones Tesis/Malla$ ns malla.tcl
num_nodos is set 25
INITIALIZE THE LIST xListHead
```

Figura 3.9. Ejecución de la simulación.

El archivo de trazas con extensión .tr contiene gran cantidad de información, con la que se puede calcular la eficiencia, adaptabilidad, tiempos de transmisión, cantidad de paquetes,

¹ Cuando se encuentra escrito en minúscula hace referencia al programa de UNIX que interpreta programas escritos en el lenguaje de programación AWK.

energía consumida, entre otros. En el anexo I se encuentra la explicación de cada uno de los campos que lo componen.

Cada fila del archivo contiene un evento diferente, puede ser un paquete enviado, recibido, perdido, entre otros. Para el análisis de la información se utiliza el lenguaje AWK, cuyo nombre deriva de las iniciales de los apellidos de sus autores. Este es un lenguaje de programación diseñado para procesar datos basados en texto. El mismo toma como entrada el archivo de trazas y permite mostrar solo la información deseada para una mejor comprensión.

3.4.3 Configuración de la simulación

En los *scripts* que se mencionan en el epígrafe anterior, los nodos deben ser configurados para crear la topología de red que se desea. Esta configuración incluye el modelo de radio propagación, el protocolo de enrutamiento, la capa MAC a utilizar, la potencia de transmisión, entre otras variables. En la tabla 3.3 se muestra la configuración empleada.

Tabla 3.3. Configuración de los nodos.

Parámetro	
Modelo de radio propagación	<i>Two Ray Ground*</i>
Capa MAC	802.15.4
Protocolo de enrutamiento	AODV
Potencia de transmisión	1 mW
Sensibilidad	-101 dBm
Consumo en transmisión	17 mA
Consumo en recepción	16 mA
Frecuencia de operación	2,4 GHz

* Esta es una versión mejorada del modelo de espacio libre, el cual tiene en cuenta el rayo directo y el rayo reflejado.

Para la simulación de las topologías también se necesita definir el tráfico de paquetes entre los nodos. El tipo de tráfico empleado es CBR (*Constant Bit Rate*), el cual envía bits a una tasa constante sin variaciones de ningún tipo. El mismo se asemeja a los datos que se van a recopilar en el diseño propuesto.

Tiempo de simulación

Este parámetro permite ver la evolución de la red desde su inicio, en el que los nodos buscan establecer conexión con el coordinador, luego se puede observar el envío de paquetes entre los nodos y la estación base. Para este efecto se simula durante 240 segundos, debido a que en la escala de tiempo de NS2 un segundo en la realidad

representa apenas dos milisegundos dentro de la simulación, lo que da un tiempo suficiente para recoger datos de la red.

Variables de análisis

Para el estudio se toman en cuenta las variables que se consideran más relevantes en análisis de eficiencia, consumo de energía, entre otras. Para dicho efecto se analizan los siguientes parámetros:

- Cantidad de paquetes enviados, recibidos, desechados y reenviados en la red.
- Retardo de los paquetes que son enviados por los nodos finales al nodo coordinador.
- Consumo total de energía de la red.
- Razón de entrega de paquetes: Relación entre la cantidad de paquetes recibidos y enviados.

3.4.4 Simulación de la topología malla

En este tipo de topología, las comunicaciones son descentralizadas y ningún nodo puede comunicarse con algún otro que no se encuentre en su rango de cobertura. Estas redes permiten a los datos saltar de nodo en nodo. De esta forma, un nodo es capaz de comunicarse con cualquier otro, de modo que los datos pueden atravesar nodos intermedios hasta llegar a su destino. La red es un sistema multi-salto, en donde todos los nodos sensores son idénticos, todos deben ser enrutadores *ZigBee*.

La topología en malla provee buena escalabilidad y flexibilidad a la red, tales como la existencia de caminos redundantes, lo que incrementa la confiabilidad de transmisión de datos extremo a extremo. Sin embargo, ya que los caminos de enrutamiento no se pueden predecir, los nodos no pueden entrar en modo de bajo consumo, lo que conduce a una pérdida de energía. Aquí radica una de las desventajas de esta topología, además, todos los dispositivos deben ser de funcionalidad completa, lo que aumenta los costos de la red.

En la figura 3.10 se muestra el resultado del archivo .nam para la topología malla. El simulador permite distinguir los nodos coordinadores de la red PAN, los enrutadores y los dispositivos finales. En este caso el coordinador está representado por el color rojo y los dispositivos de funcionalidad completa de color azul.

Para la obtención de los datos se aplica el filtro AWK al archivo de trazas resultante de la simulación. En la tabla 3.5 se muestran los resultados de esta simulación luego de filtrados

los datos. En la misma puede observarse el gran consumo de energía de la red, un factor que es sumamente importante en las redes de sensores inalámbricos.

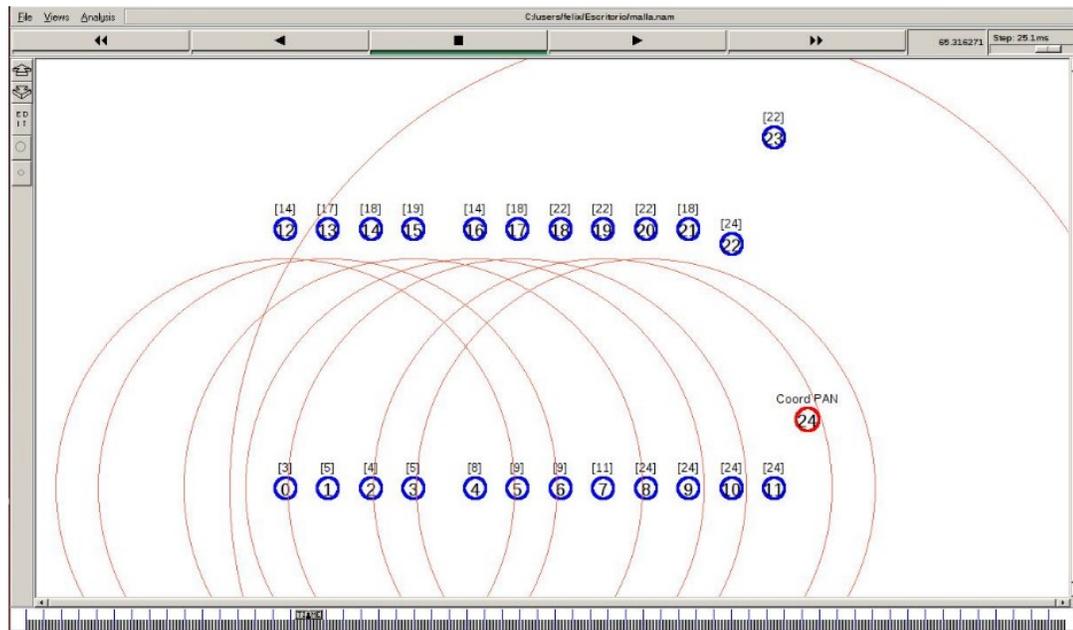


Figura 3.10. Simulación de la topología en malla.

Tabla 3.4. Estadísticas de la topología malla.

Total de paquetes enviados	3484
Total de paquetes recibidos	3113
Total de paquetes desechados	346
Total de paquetes reenviados	6184
Razón de entrega de paquetes (%)	89
Latencia promedio (ms)	0,0879483
Consumo total de energía (J)	18.529.903

3.4.5 Simulación de la topología *cluster-tree*

La topología *cluster-tree* combina los beneficios de las topologías anteriores, como son buena escalabilidad, sincronización, eficiencia energética, lo cual es más apropiado para aplicaciones de media escala, al tiempo que utilizan nodos alimentados con baterías. Esta topología organiza los nodos en grupos lógicos llamados *cluster*. La misma permite el uso del modo *beacon* habilitado, con el cual se logra un mayor ahorro energético.

Cada enrutador, incluyendo el coordinador de la red, forma un *cluster*, el cual es conocido como cabeza de *cluster*. Todos los nodos asociados con un determinado cabeza de *cluster* pertenecen a ese grupo, y todas sus transmisiones son controladas por el cabeza de *cluster*.

Cada grupo puede ser visto como una subred en estrella. El coordinador de la red PAN es identificado como la raíz del árbol y forma el *cluster* inicial. Los otros enrutadores a su vez se unen al árbol, estableciéndose como cabezas de *cluster* e inicializándose para generar las tramas *beacons*.

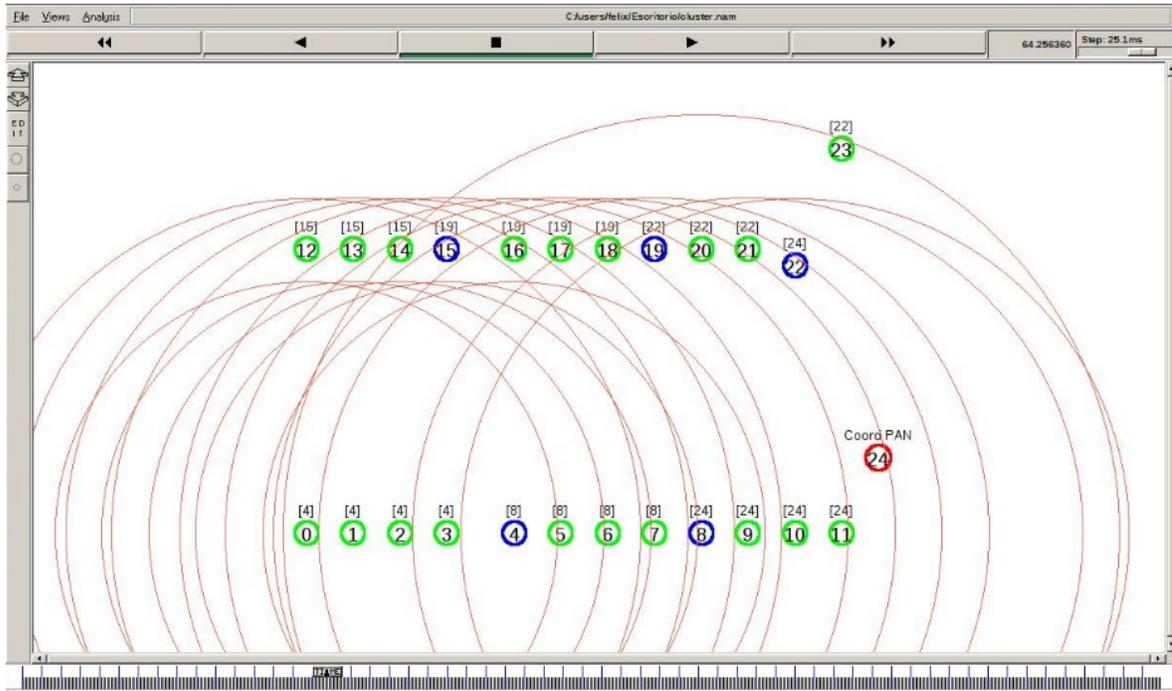


Figura 3.11. Simulación de la topología *cluster-tree*.

Contrario a la topología en malla, hay un solo camino de enrutamiento entre cualquier par de nodos. Por lo tanto, la comunicación es eficiente en tiempo, porque cada nodo solo interactúa con su predefinido conjunto de nodos cercanos. Además, los nodos conocen su tiempo activo con anticipación, por lo tanto, cada nodo puede ahorrar su energía entrando al modo de baja potencia cuando no esté participando en el enrutamiento. Esta topología permite que solo los enrutadores y el coordinador de la red sean dispositivos de funcionalidad completa, mientras que los nodos finales pueden ser de funcionalidad reducida, lo que abarata el costo de la instalación.

Para la simulación de la topología *cluster-tree* se seleccionan 5 nodos como cabezas de *cluster* y un coordinador PAN. Todos los demás nodos son dispositivos de funcionalidad reducida. En la figura 3.11 se observa la topología, en la que el coordinador PAN está representado por el color rojo, los cabezas de *cluster* con el color azul y los dispositivos finales de color verde.

Para el caso de esta topología, se varía el parámetro BO descrito en el capítulo 1, mientras que la variable SO se mantiene constante en un valor de tres. El incremento de este factor es directamente proporcional al ahorro de energía, pero puede influir negativamente en la latencia de las transmisiones. En la aplicación que se simula en este caso no ocurre de esta manera, pues el sensado se realiza cada diez minutos, sin embargo, en otras aplicaciones, donde los requerimientos de sensado sean mayores, es necesario tener en cuenta el aumento de esta variable.

Tabla 3.5. Estadísticas de la topología cluster-tree.

Valor de BO	3	4	5	6
Razón de entrega de paquetes (%)	95	97	97	98
Consumo total de energía (J)	8,29	7,60	7,08	7,01
Latencia promedio (ms)	0,09	0,08	0,09	0,03

En la tabla 3.5 se muestran los resultados obtenidos de la simulación, luego de ser filtrados con el *script* realizado en el lenguaje AWK. En la tabla se puede observar que a medida que aumenta el parámetro BO, el consumo de energía disminuye. Esta situación se debe a que los coordinadores envían con menor frecuencia las tramas balizas, y por tanto se logra un ahorro energético.

De las cuatro simulaciones que se realizan, se aprecia que la que ofrece las mejores estadísticas es la última, por lo que se procede a comparar esta con los resultados obtenidos en la simulación de la topología en malla. En la figura 3.9 se observa dicha comparación.

Luego de analizar las dos topologías, se puede concluir que la idónea para el caso en particular de la “Casa de Cultivos Protegidos San José”, es la topología *cluster-tree*, ya que la misma presenta mejor razón de entrega de paquetes, además proporciona un mayor ahorro energético, factor de vital importancia en las redes de sensores inalámbricos. Esta topología permite además, que solo se tengan 6 dispositivos de funcionalidad completa, mientras que el resto pueden ser dispositivos de funcionalidad reducida, lo que abarata el costo total de la instalación.

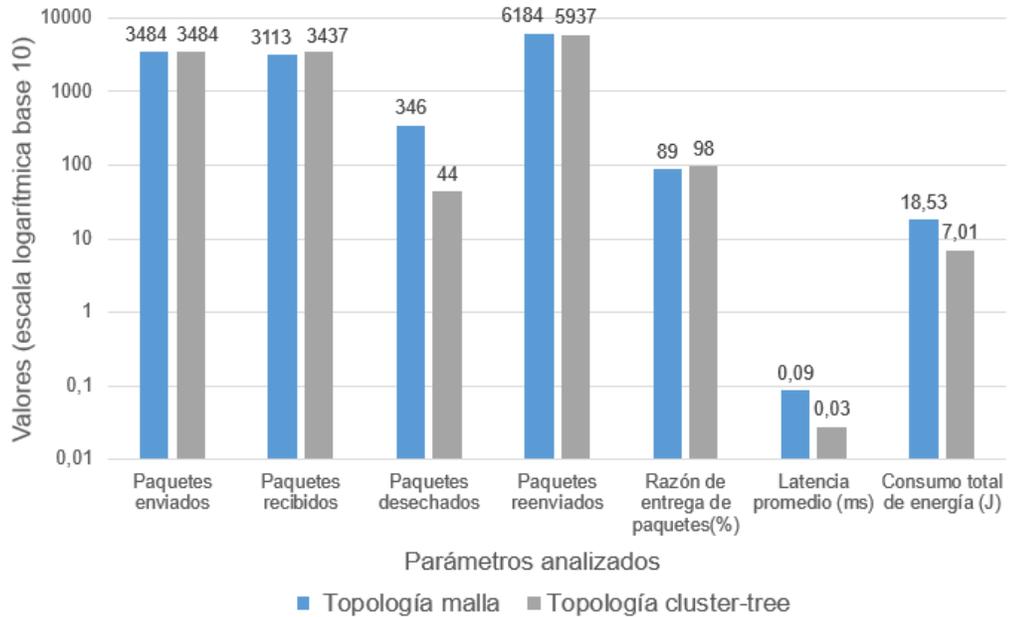


Figura 3.12. Comparación de las topologías.

En la figura 3.13 se presenta el diseño final de la red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”. En la misma, el punto simbolizado por el color rojo representa el coordinador de la red PAN, los nodos de color azul constituyen nodos enrutadores y los de color verde representan nodos finales. Como se aprecia en la figura, se conforman un total de 6 clusters.

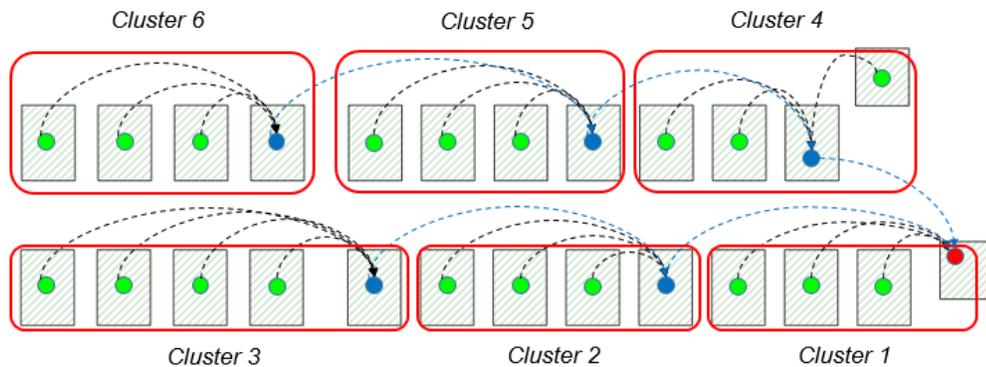


Figura 3.13. Esquema del diseño propuesto.

3.5 Valoración económica

Para adoptar esta solución se requiere una valoración económica que permita deducir si es viable o no esta propuesta. El criterio de valoración que se emplea para este proyecto es el método dinámico Valor Actual Neto (VAN).

El Valor Neto Actualizado o Valor Actual Neto (VAN) de un proyecto mide en dinero corriente el grado de mayor riqueza que tendrá el inversionista en el futuro si emprende el proyecto. Se define como el valor actualizado del flujo de ingresos netos obtenidos durante la vida útil económica del proyecto a partir de la determinación por año de las entradas y salidas de divisas en efectivo, desde que se incurre en el primer gasto de inversión durante el proceso inversionista hasta que concluyen los años de operación o funcionamiento de la inversión.

$$VAN = (FC_0 * a_0) + (FC_1 * a_1) + \dots + (FC_j * a_j) + \dots + (FC_n * a_n) \quad (3.1)$$

En donde:

- FC es la corriente de liquidez neta de un proyecto, o ingreso neto, positivo o negativo, que se obtiene en los años 0, 1, 2, ,3, ..., n.
- a es el factor de actualización en los años 1, 2, 3,..., n, correspondiente a la tasa de que se utilice.

Tabla 3.6. Costos de la instalación.

Componente	Unidad	Cantidad	Precio (CUC)	Importe (CUC)
EN2100	U	19	350	6650
EN2120	U	5	450	2250
eS1401	U	24	136	3264
eS1101	U	24	244	5856
eB2110	U	1	350	350
eG2100	U	1	400	400
Importe total				18770

Se parte del año cero porque se consideran desde los primeros gastos de inversión, es decir el análisis se realiza a partir del período de construcción. Es conveniente anotar que la tasa de descuento puede cambiar de año en año. El factor de actualización se puede obtener de las tablas de actualización editadas por organismos internacionales como la ONUDI. En este caso el factor de actualización se toma del 7,5 %. En la tabla 3.6 se listan los costos de la inversión del proyecto propuesto.

Para el cálculo del VAN, se hace uso de la herramienta Microsoft Excel, la cual tiene implementada esta función. Se necesita conocer los ingresos de la instalación en el período de un año para computar este criterio de evaluación. En entrevista con la Técnica Integral

de la “Casa de Cultivos Protegidos San José”, se adquieren las ganancias netas del año 2014, las cuales son de 27354 CUC. El análisis se realiza para tres años².

En el programa Excel, se anota en una celda la tasa de descuento y se posiciona en la celda en la que se desea que aparezca el resultado del VAN. En el menú de funciones se selecciona la función VNA. En el cuadro tasa se introduce la tasa de actualización y en el cuadro valor 1, se introducen las celdas que contienen los flujos, sin incluir la inversión. Luego se selecciona la celda del VAN y se suma a la celda de la inversión, y de esta forma se obtiene el valor actual neto del proyecto. En la tabla 3.7 se muestran los resultados. Como se puede apreciar el VAN es positivo, por lo tanto el proyecto se acepta.

Tabla 3.6. Cálculo del VAN.

Inversión	-18770
Año 1	27354
Año 2	27354
Año 3	27354
Tasa de descuento	7,50%
VAN	52.364,78

El capítulo se dedica al diseño de la red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”. Para determinar la topología de la red se realizan las simulaciones en NS2 de las topologías *cluster-tree* y malla, con el objetivo de compararlas y analizar cuál ofrece el mejor desempeño. Para el análisis de los resultados se programa un filtro con el lenguaje AWK, ya que la herramienta que posee el NS2 no es compatible con las trazas generadas en las simulaciones de redes de sensores inalámbricos.

² El valor de las ganancias se asume igual para los tres años, esto es solo un aproximado, pues estas ganancias están sujetas a cambio de un período a otro.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- El uso del estándar de comunicación *ZigBee* en una red de sensores inalámbricos resulta idóneo, porque cumple con los requerimientos de ahorro energético sin afectar el funcionamiento de la red.
- La plataforma *eKo* resulta la adecuada para ser utilizada en el proyecto propuesto, debido a que presenta la robustez necesaria para ser desplegada en entornos agrícola, además de presentar un bajo consumo energético.
- Se diseñó una red de sensores inalámbricos para la “Casa de Cultivos Protegidos San José”, la cual consta de 25 nodos distribuidos en 23 casas de cultivo, una de posturas y un nodo coordinador ubicado en la estación base, los cuales permiten la supervisión de temperatura, humedad relativa y radiación solar.
- Para el análisis de los resultados de las simulaciones se programó un filtro empleando el lenguaje AWK, lo cual permitió obtener solo la información deseada a la hora de realizar la comparación de las topologías.
- Se simularon las topologías malla y *cluster-tree*, utilizando el simulador NS2, y se comprobó que esta última es la adecuada para emplear en el diseño propuesto, ya que presenta mejor razón de entrega de paquetes y, particularmente, un bajo consumo de energía.

Recomendaciones

- Realizar estudios con otras plataformas, como la plataforma Arduino, para mejorar los costos presentados en el presente proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. V. Capella Hernández, *Redes inalámbricas de sensores: Una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos.*, Universidad Politécnica de Valencia, Abril 2010.
- [2] R. Fernández Martínez, J. Ordieres Meré, F. J. Martínez, A. González Marco, F. Alba Elías y R. Lostado Lorza, «Redes inalámbricas de sensores: teoría y aplicación práctica.» Universidad de la Rioja, 2009.
- [3] N. Islam y . Z. Ahmed Shaikh, *A review of wireless sensors and networks' applications in agriculture.*, Elsevier, 2011.
- [4] C. A. Castillo Luzón, «Implementación de un prototipo de red de sensores inalámbricos para invernaderos.» Quito, 2007.
- [5] [En línea]. Available: <http://www.wisevine.info>. [Último acceso: enero 2014].
- [6] L. Gang, «Design of greenhouse environment monitoring and controlling system based on bluetooth technology.» Springer, China, 2013.
- [7] C. Wang, C. Zhao, X. Qiao y Y. Zhang, «The design of wireless sensor networks node for measuring the greenhouse's environment parameters.» Springer, Boston, 2014.
- [8] A. Casanova, O. Gómez, T. Depestre, A. Igarza y M. León, *Guía Técnica para la producción protegida de hortalizas en casas de cultivos tropical con efecto sombrilla.*, La Habana.
- [9] G. Olimpia, A. Casanova y H. Laterrot, *Mejora genética y manejo del cultivo del tomate para la producción en el Caribe.*, La Habana, 2000.
- [10] M. Largo Medero, E. A. Medina Hernández y G. Fumero Pecosó, *Sostenibilidad económica de los cultivos protegidos.*, Villa Clara, 2014.
- [11] A. S. González, Interviewee, *Técnica Integral*. [Entrevista]. 14 abril 2015.
- [12] R. Jaimez, «Casas de cultivo: estrategia de producción sustentable.» Buenos Aires, 2014.
- [13] A. S. González, Interviewee, *Técnica Integral*. [Entrevista]. 14 abril 2015.
- [14] P. RENTALA, R. MUSUNNURI, S. GANDHAM y U. SAXENA, *Survey on Sensor Networks*, 2001.
- [15] K. H y W. A, *A short survey of wireless sensor networks*, Berlín, 2003.
- [16] J. YICK, B. MUKHERJEE y D. GHOSAL, *Wireless sensor network survey.*, 2008.

- [17] L. F. Smart Enviroments: Technologies, Protocols and Applications., 2004, New York.
- [18] J. . A. López Riquelme, «Contribución a las redes de sensores inalámbricas. Estudio e implementación de soluciones hardware para agricultura de precisión.,» Cartagena, 2011.
- [19] K. Holger y A. Willig, Protocols and architectures of wireless sensor networks., Springer, 2005.
- [20] M. C. Acoste Ponce, «Estudio del estándar IEEE 802.15.4 ZigBee para comunicaciones inalámbricas de área personal de bajo consumo de energía y su comparación con el estándar IEEE 802.15.1 Bluetooth.,» Quito, 2006.
- [21] J. F. Reyes Coellar y E. . M. Tene Guamán, «Diseño de la arquitectura para transmisión de datos de los sensores de campo del PROMAS.,» Cuenca, 2014.
- [22] J. Martín Moreno y D. Ruiz Fernández, «Informe Técnico: Protocolo ZigBee (IEEE 802.15.4).,» 2007.
- [23] [En línea]. Available: http://www.jennic.com/files/support_files/JN-UG-3024-IEEE802.15.4-1v1.pdf. [Último acceso: enero 2014].
- [24] P. Levis, S. Madden, K. Whitehouse y A. Woo, «TinyOs: An Operating System for Sensor Networks.».
- [25] A. M. Ortiz Torres, Técnicas de enrutamiento inteligentes para redes de sensores inalámbricas., La Mancha, 2011.
- [26] The NS2 manual., 2011.
- [27] [En línea]. Available: <http://www.omnetpp.org/>. [Último acceso: enero 2014].
- [28] [En línea]. Available: <http://nsr.bioeng.washington.edu/jsim>. [Último acceso: enero 2014].
- [29] S. R. Maroto Cantillo, «Desarrollo de aplicaciones basadas en WSN.,» Valencia, 2010.
- [30] Z. Shaikh, «Crop irrigation control using wireless sensor and actuator network (wsan).,» 2010.
- [31] Y. Zhou , « A wireless design of low-cost irrigation system using ZigBee technology.,» 2009.
- [32] A. Jiménez, D. Ravelo y J. Gómez, «Sistema de adquisición, almacenamiento y análisis de información fenológica para el manejo de plagas y enfermedades de un duraznero mediante tecnologías de agricultura de precisión.,» *Tecnura*, vol. 14, n° 27, pp. 41-51, 2010.
- [33] M. E. Gordon y B. S. Vázquez , Análisis de desempeño de redes de sensores inalámbricos en tiempo real aplicadas a monitorización volcánica., Ecuador, 2012.

- [34] E. E. Flores Carbajal, Redes de sensores inalámbricas aplicadas a la medicina, 2012.
- [35] «Micaz datasheet.».
- [36] «Crossbow,» [En línea]. Available: www.xbow.com.
- [37] «MTS-MDA Series User Manual.,» 2014.
- [38] «Mica2 datasheet.».
- [39] «Mica2Dot datasheet.».
- [40] «Imote2 datasheet.».
- [41] «Iris datasheet».
- [42] «Tmote Sky datasheet.».
- [43] «eKo Datasheet,» 2015.
- [44] «Libelium,» [En línea]. Available: www.libelium.com.
- [45] «Waspote datasheet.,» 2015.
- [46] « Control del medio ambiente,» [En línea]. Available: <http://www.fao.org/docrep/005/s8630s/s8630s06.htm#bm06>. [Último acceso: marzo 2015].
- [47] «Infoagro,» [En línea]. Available: http://www.infoagro.com/industria_auxiliar/control_climatico.htm. [Último acceso: marzo 2015].
- [48] R. Morais, B. Cunha, M. Cordeiro, C. Serodio y P. Salgado, «Solar data acquisition wireless network for agricultural applications.,» de *The 19th IEEE Convention of Electrical and Electronics Engineers in Israel*, Israel, 2012.
- [49] Weston, Fundamentos de administración financiera., Décima Edición..

GLOSARIO DE TÉRMINOS

AODV: *Ad hoc On-Demand Distance Vector.*

APS: *Application Support Sub layer* (Sub capa de Soporte de Aplicación).

BI: *Beacon Interval* (Intervalo de la trama baliza).

BO: *MacBeaconOrder* (Orden de la trama baliza).

CAP: *Contention Access Period.* (Período de Acceso por Contención).

CBR: *Constant Bit Rate* (Tasa de bits constante)

CFP: *Contention Free Period* (Período libre de contención).

CPU: *Central Processing Unit* (Unidad de Procesamiento Central).

CSMA/CA: *Carrier Sense Multiple Access with Collision Avoidance* (Acceso múltiple con detección de portadora y evasión de colisiones).

DSDV: *Destination-Sequenced Distance Vector.*

DSR: *Dynamic Source Routing.*

E/S: Entrada/Salida.

EPROM: *Erasable Programmable Read-Only Memory* (Memoria de solo lectura programmable borrrable).

ESB: *Environmental Sensor Bus.*

FFD: *Full Function Device.* (Dispositivo de funcionalidad completa)

GPRS: *General Packet Radio Service* (Servicio General de Paquetes vía radio).

GPS: *Global Position System.* (Sistema de Posicionamiento Global).

GSM: *Global System for Mobile communications* (Sistema Global para las Comunicaciones Móviles).

GUI: *Graphical User Interface.* (Interfaz Gráfica de Usuario).

I2C: *Inter-Integrated Circuit* (Bus de comunicación serie).

IEEE: *Institute of Electrical and Electronics Engineers* (Instituto de Ingenieros Eléctricos Electrónicos).

IP: *Internet Protocol*. (Protocolo de Internet).

ISM: *Industrial, scientific and medical* (Banda de frecuencias reservadas internacionalmente para uso no comercial en áreas industrial, científica y médica).

J-Sim: *Java Simulator*.

LR-WPAN: *Low Rate Wireless Personal Area Network* (Redes inalámbricas de área personal con bajas tasas de transmisión de datos).

MAC: *Medium Access Control*. (Control de acceso al medio).

MANTIS: *Multimodal system for NeTworks of In-situ wireless Sensors*.

NAM: *Network Animator*.

NS2: *Network Simulator 2*.

Omnet++: *Objective Modular Network Testbed in C++*.

PAN ID: *Personal Area Network Identification* (Identificador de la red de área personal).

PAN: *Personal Area Network*. (Red de área personal).

PDA: *Personal Digital Assistance* (Asistente personal digital).

RAM: *Random Access Memory*. (Memoria de acceso aleatorio).

RFD: *Reduced Function Device*. (Dispositivo de funcionalidad reducida).

RIP: *Routing Information Protocol*.

RISC: *Reduced Instruction Set Computer* (Computadora con conjunto de instrucciones reducidas).

SD: *Superframe duration*. (Duración de la supertrama).

SO: *MacSuperframeOrder*. (Orden de la supertrama).

SRD: *Short Range Device*.

SRAM: *Static Random Access Memory* (Memoria estática de acceso aleatorio).

UART: *Universal Asynchronous Receiver-Transmitter* (Transmisor-Receptor Asíncrono Universal).

UNII: *Unlicensed National Information Infrastructure.*

USB: *Universal Serial Bus* (Bus Universal en Serie).

VAN: Valor Actual Neto.

WSN: *Wireless Sensor Network.* (Red de sensores inalámbricos).

ZC: *ZigBee Coordinator.* (Coordinador *ZigBee*).

ZDO: *ZigBee Device Object.* (Objeto de dispositivo *ZigBee*).

ZED: *ZigBee End Device.* (Dispositivo final *ZigBee*).

ZR: *ZigBee Router.* (Enrutador *ZigBee*).

ZRP: *Zone Routing Protocol.*

-	-M...	Los campos que empiecen con esta marca se refieren a información acerca de los paquetes a nivel de capa MAC.
12	-Ma	Tiempo esperado en segundos para enviar el paquete a través del canal inalámbrico.
13	-Md	Dirección MAC del nodo destino.
14	-Ms	Dirección MAC del nodo fuente.
15	-Mt	Tipo de capa MAC
-	-I...	Los campos que empiecen con esta marca se refieren a información acerca de los paquetes a nivel de IP.
16	-Is	Dirección IP del nodo fuente seguido del número de puerto del nodo fuente.
17	-Id	Dirección IP del nodo destino seguido del número de puerto del nodo destino.
18	-It	Tipo de paquete: CBR, DSR, ACK, UDP, TCP, ARP,...
19	-Il	Tamaño del paquete en bytes
20	-If	Identificador
21	-Ii	Identificador de secuencia global del paquete. Identifica el paquete en el fichero de trazas.
22	-Iv	Valor del TTL
-	-P...	Los campos que empiecen con esta marca se refieren a información de los paquetes a nivel de aplicación.
23	-Pn	<p>En este campo se indica el tipo de aplicación:</p> <ol style="list-style-type: none"> 1. ARP: Address Resolution Protocol 2. <i>DSR: Dynamic Source Routing.</i> 3. <i>CBR: Constant Bit Rate.</i> <p>En función de este campo, el número y tipo de campos siguientes son distintos. Este ejemplo está referido para un paquete CBR.</p>
24	-Pi	Número de secuencia CBR.
25	-Pf	Número de veces que el paquete ha sido reenviado.
26	-Po	Número óptimo de reenvíos necesarios.

Anexo II. Implementación de la topología *cluster-tree*.

```
#####
#           Topología cluster-tree utilizando el estándar ZigBee           #
#-----#
#                               Universidad de Oriente                               #
#                               Facultad de Ingeniería Eléctrica                       #
#                               Departamento de Telecomunicaciones                     #
#####

#=====
#                               Definición de las opciones                               #
#=====

set val(canal)      Channel/WirelessChannel      ;# Tipo de canal
set val(prop)      Propagation/TwoRayGround     ;# Modelo de radio-propagación
set val(capa)      Phy/WirelessPhy/802_15_4     ;# Capa física
set val(mac)       Mac/802_15_4                 ;# Capa MAC
set val(coola)     Queue/DropTail/PriQueue      ;# Tipo de interfaz de cola
set val(ll)        LL                           ;# Enlace lógico
set val(ant)       Antenna/OmniAntenna         ;# Modelo de antena
set val(tcoola)    150                          ;# Tamaño de la cola
set val(nn)        25                          ;# Número de nodos
set val(penr)      AODV                         ;# Protocolo de enrutamiento
set val(x)         195                          ;# Dimensiones
set val(y)         160                          ;# Dimensiones
set val(nam)       cluster.nam                  ;# Archivo nam
set val(trafico)   cbr                          ;# Tipo de trafico

#=====
#                               Definición de los tiempos                               #
#=====

set Tiempo1      55.0
set Tiempo2      56.0
set Tiempo3      57.0
set Tiempo4      58.0
set Tiempo5      59.0
set Tiempo6      60.0
set Tiempo7      61.0
set Tiempo8      62.0
```

```

set Tiempo9      63.0
set Tiempo10     64.0
set Tiempo11     65.0
set Tiempo12     66.0
set Tiempo13     67.0
set Tiempo14     68.0
set Tiempo15     69.0
set Tiempo16     70.0
set Tiempo17     71.0
set Tiempo18     72.0
set Tiempo19     73.0
set Tiempo20     74.0
set Tiempo21     75.0
set Tiempo22     76.0
set Tiempo23     77.0
set Tiempo24     78.0
set Parada       240

```

```

#=====
#                               Definición de las variables globales
#=====

set ns          [new Simulator]
$ns            use-newtrace
set tracefd     [open ./cluster.tr w]
$ns trace-all $tracefd
if { "$val(nam)" == "cluster.nam" } {
    set namtrace [open ./$val(nam) w]
    $ns namtrace-all-wireless $namtrace $val(x) $val(y)
}
$ns puts-nam-traceall {# nam4wpan #}
Mac/802_15_4 wpanCmd verbose on
Mac/802_15_4 wpanNam namStatus on
Mac/802_15_4 wpanNam PANCoorClr red
Mac/802_15_4 wpanNam CoorClr blue
Mac/802_15_4 wpanNam DevClr Green

```

```

=====
#
#           Definición de los parámetros de la capa física
#
=====
Phy/WirelessPhy set CStresh_ 3.98107e-13
Phy/WirelessPhy set RXThresh_ 3.98107e-13
Phy/WirelessPhy set CPThresh_ 10
Phy/WirelessPhy set Pt_ 0.001
Phy/WirelessPhy set freq_ 2.4e+9

=====
#
#           Definición de los parámetros de la antena
#
=====
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.3
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

=====
#
#           Creación del objeto topografía
#
=====
set topologia [new Topography]
$topologia load_flatgrid $val(x) $val(y)
set god [create-god $val(nn)]
set canal_1 [new $val(canal)]

=====
#
#           Configuración de los nodos
#
=====
$ns node-config -adhocRouting $val(penr) \
    -lType $val(l) \
    -macType $val(mac) \
    -ifqType $val(cola) \
    -ifqLen $val(tcola) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(capa) \
    -topoInstance $topologia \
    -agentTrace ON \

```

```

-routerTrace ON \
-macTrace OFF \
-movementTrace OFF \
-energyModel "EnergyModel" \
-initialEnergy 5760 \
-rxPower 16e-3 \
-txPower 17e-3 \
-idlePower 712e-6 \
-sleepPower 144e-9 \
-channel $canal_1 \

#=====
#                               Creación de los nodos
#=====
for {set i 0} {$i < $val(nn)} {incr i} {
    set nodos($i) [$ns node]
    $nodos($i) random-motion 0
}
source ./nodos.scn3

#=====
#                               Inicialización de la red PAN
#=====
$ns at 0.0 "$nodos(24) NodeLabel Coord PAN"
$ns at 0.0 "$nodos(24) sscs startCTPANCoord 1 6 3"
$ns at 1.0 "$nodos(11) sscs startCTDevice 0 0"
$ns at 2.0 "$nodos(10) sscs startCTDevice 0 0"
$ns at 3.0 "$nodos(9) sscs startCTDevice 0 0"
$ns at 4.0 "$nodos(8) sscs startCTDevice 1 1 1 6 3"
$ns at 6.0 "$nodos(7) sscs startCTDevice 0 0"
$ns at 7.0 "$nodos(6) sscs startCTDevice 0 0"
$ns at 8.0 "$nodos(5) sscs startCTDevice 0 0"
$ns at 9.0 "$nodos(4) sscs startCTDevice 1 1 1 6 3"
$ns at 10.0 "$nodos(3) sscs startCTDevice 0 0"

```

³ Esta línea de código hace referencia a otro archivo donde se encuentran las ubicaciones de cada uno de los nodos, el cual no se incluye debido a su extensión.

```

$ns at 11.0 "$nodos(2) sscs startCTDevice 0 0"
$ns at 12.0 "$nodos(1) sscs startCTDevice 0 0"
$ns at 13.0 "$nodos(0) sscs startCTDevice 0 0"
$ns at 16.0 "$nodos(22) sscs startCTDevice 1 1 1 6 3"
$ns at 17.0 "$nodos(23) sscs startCTDevice 0 0"
$ns at 18.0 "$nodos(21) sscs startCTDevice 0 0"
$ns at 19.0 "$nodos(20) sscs startCTDevice 0 0"
$ns at 21.0 "$nodos(19) sscs startCTDevice 1 1 1 6 3"
$ns at 23.0 "$nodos(18) sscs startCTDevice 0 0"
$ns at 23.0 "$nodos(17) sscs startCTDevice 0 0"
$ns at 24.0 "$nodos(16) sscs startCTDevice 0 0"
$ns at 26.0 "$nodos(15) sscs startCTDevice 1 1 1 6 3"
$ns at 27.0 "$nodos(14) sscs startCTDevice 0 0"
$ns at 28.0 "$nodos(13) sscs startCTDevice 0 0"
$ns at 29.0 "$nodos(12) sscs startCTDevice 0 0"

```

Mac/802_15_4 wpanNam PlaybackRate 3ms

```
$ns at $Tiempo1 "puts "\nTransmitiendo datos ... \n\""
```

```

#=====
#                               Tráfico entre los nodos
#=====

```

```

proc cbrtrafico { origen destino intervalo inicio } {
    global ns nodos
    set udp_($origen) [new Agent/UDP]
    eval $ns attach-agent \ $nodos($origen) \ $udp_($origen)
    set null_($destino) [new Agent/Null]
    eval $ns attach-agent \ $nodos($destino) \ $null_($destino)
    set cbr_($origen) [new Application/Traffic/CBR]
    eval \ $cbr_($origen) set packetSize_ 5
    eval \ $cbr_($origen) set interval_ $intervalo
    eval \ $cbr_($origen) attach-agent \ $udp_($origen)
    eval $ns connect \ $udp_($origen) \ $null_($destino)
    $ns at $inicio "$cbr_($origen) start"
}
puts "\nTráfico: $val(trafico)"

```

\$ns at \$Tiempo1 "Mac/802_15_4 wpanNam PlaybackRate 0.5ms"

```
$val(trafico)trafico 0 24 1.2 $Tiempo1
$val(trafico)trafico 1 24 1.2 $Tiempo2
$val(trafico)trafico 2 24 1.2 $Tiempo3
$val(trafico)trafico 3 24 1.2 $Tiempo4
$val(trafico)trafico 4 24 1.2 $Tiempo5
$val(trafico)trafico 5 24 1.2 $Tiempo6
$val(trafico)trafico 6 24 1.2 $Tiempo7
$val(trafico)trafico 7 24 1.2 $Tiempo8
$val(trafico)trafico 8 24 1.2 $Tiempo9
$val(trafico)trafico 9 24 1.2 $Tiempo10
$val(trafico)trafico 10 24 1.2 $Tiempo11
$val(trafico)trafico 11 24 1.2 $Tiempo12
$val(trafico)trafico 12 24 1.2 $Tiempo13
$val(trafico)trafico 13 24 1.2 $Tiempo14
$val(trafico)trafico 14 24 1.2 $Tiempo15
$val(trafico)trafico 15 24 1.2 $Tiempo16
$val(trafico)trafico 16 24 1.2 $Tiempo17
$val(trafico)trafico 17 24 1.2 $Tiempo18
$val(trafico)trafico 18 24 1.2 $Tiempo19
$val(trafico)trafico 19 24 1.2 $Tiempo20
$val(trafico)trafico 20 24 1.2 $Tiempo21
$val(trafico)trafico 21 24 1.2 $Tiempo22
$val(trafico)trafico 22 24 1.2 $Tiempo23
$val(trafico)trafico 23 24 1.2 $Tiempo24
```

Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato

Mac/802_15_4 wpanNam FlowClr -p ARP -c green

Mac/802_15_4 wpanNam FlowClr -p MAC -s 0 -d -1 -c navy

```
#=====
#           Tamaño de los nodos en el NAM y configuración del fin de simulación
#=====
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $nodos($i) 7
}
```

```
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns at $Parada "$nodos($i) reset";
}
proc stop {} {
    global ns tracefd Tiempo1 val env
    $ns flush-trace
    close $tracefd
    set hasDISPLAY 0
    foreach index [array names env] {
        if {("$index" == "DISPLAY") && ("$env($index)" != "")} {
            set hasDISPLAY 1
        }
    }
    if {("$val(nam)" == "cluster.nam") && ("$hasDISPLAY" == "1")} {
    }
}
puts "\nComenzando simulación..."
$ns at $Parada "stop"
$ns at $Parada "puts \"NS Saliendo...\n\""
$ns at $Parada "$ns halt"
$ns run
```

Anexo III. Código del filtro AWK para el análisis de los resultados de las simulaciones.

```
BEGIN {
print("\n\n***** Estadísticas de la topología cluster-tree*****\n");
paq_env[25] = 0;
paq_des[25] = 0;
paq_recb[25] = 0;
paq_re[25] = 0;
energia[25] = 5760;
total_paq_env=0;
total_paq_rec=0;
total_paq_des=0;
total_paq_ree=0;
raz_paq_ent = 0;
total_hop_count = 0;
avg_hop_count = 0;
overhead = 0;
start = 0.000000000;
end = 0.000000000;
duracion_paq = 0.000000000;
recvnum = 0;
delay = 0.000000000;
sum = 0.000000000;
i=0;
total_energia_cons = 0.000000;
}
{
evento= $1;
tiempo= $3;
num_nodo = $5;
nivel_energia = $7;
id_nodo = $9;
nivel= $19;
tipo_paq = $35;
id_paq = $41;
```

```
no_de_reenvios= $49;
# Conteo de los paquetes en la red
if((tipo_paq == "cbr") && (evento == "s") && (nivel=="AGT")) {
for(i=0;i<25;i++) {
if(i == id_nodo) {
paq_env[i] = paq_env[i] + 1; }
}}
else if((tipo_paq == "cbr") && (evento == "r") && (level=="AGT")) {
for(i=0;i<25;i++) {
if(i == id_nodo) {
paq_recb[i] = paq_recb[i] + 1; }
}}
else if((tipo_paq == "cbr") && (evento == "d")) {
for(i=0;i<25;i++) {
if(i == id_nodo) {
paq_des[i] = paq_des[i] + 1; }
}}
else if((tipo_paq == "cbr") && (evento == "f")) {
for(i=0;i<25;i++) {
if(i == id_nodo) {
paq_re[i] = paq_re[i] + 1; }
}}
# Cálculo del promedio End to End Delay
if (( evento == "s") && (tipo_paq == "cbr" ) && ( nivel == "AGT" )) { start_time[id_paq] = tiempo; }
if (( state == "r") && ( pkt_type == "cbr" ) && ( level == "AGT" )) { end_time[id_paq] = tiempo;}
else { end_time[id_paq] = -1; }
if(state == "N") {
for(i=0;i<25;i++) {
if(i == num_nodo) {
energia[i] = energia[i] - (energia[i] - nivel_energia);
}}}}
END {
for(i=0;i<25;i++) {
```

```
#Se guarda la información de los paquetes de cada nodo en archivos .txt
printf("%d %d \n",i, paq_env[i]) > "paqenv.txt";
printf("%d %d \n",i, paq_recb[i]) > "paqrecib.txt";
printf("%d %d \n",i, paq_des[i]) > "paqdes.txt";
printf("%d %d \n",i, paq_re[i]) > "paqre.txt";
printf("%d %.6f \n",i, energia[i]) > "energiare.txt";

total_paq_env = total_paq_env + paq_env[i];
total_paq_rec = total_paq_rec + paq_recb[i];
total_paq_des = total_paq_des + paq_des[i];
total_paq_ree = total_paq_ree + paq_re[i];
total_energia_cons = total_energia_cons + energia[i];
}

printf("Total Pquetes Enviados: %d\n",total_paq_env);
printf("Total Packets Recibidos: %d\n",total_paq_rec);
printf("Total Packets Desechados : %d\n",total_paq_des);
printf("Total Packets Reenviados : %d\n", total_paq_ree);
raz_paq_ent = (total_paq_rec/total_paq_env)*100;
printf("Razón de entrega de paquetes: %d\n", raz_paq_ent);

for ( i in end_time ) {
start = start_time[i];
end = end_time[i];
duracion_paq = end - start;
if ( duracion_paq > 0 ) { sum += duracion_paq; recvnum++; }
}
delay=sum/recvnum;
printf("Latencia promedio: %.9f ms\n", delay);
printf("Consumo total de energía: %.6f\n", (25*5760)-total_energia_cons);
}
```