

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería Electrónica**



**Diseño de simulador para redes inalámbricas de sensores con nodos  
CRTECMOTE**

**Informe de Proyecto de Graduación para optar por el título de Ingeniería  
Electrónica con el grado académico de Licenciatura**

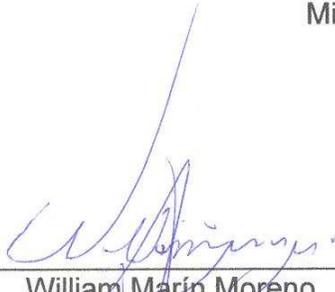
**Laura Cristina Cabrera Quirós**

**Cartago, noviembre 2011**

**INSTITUTO TECNOLOGICO DE COSTA RICA  
ESCUELA DE INGENIERIA ELECTRONICA  
PROYECTO DE GRADUACIÓN  
TRIBUNAL EVALUADOR**

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

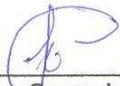
**Miembros del Tribunal**



Ing. William Marín Moreno  
Profesor lector



Ing. Carlos Meza Benavides PhD  
Profesor lector



Ing. Johan Carvajal Godínez  
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, noviembre 2011

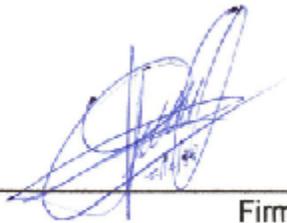
## **Declaratoria de autenticidad**

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo realizado y por el contenido del correspondiente anteproyecto.

Cartago, noviembre 2011



---

Firma del autor  
Laura Cristina Cabrera Quirós  
Céd. 1-1343-0740

## Resumen

Este documento pretende dar una descripción detallada del proceso de reestructuración realizada a la herramienta de simulación Castalia para adaptarla al funcionamiento de las redes inalámbricas de sensores con nodos CRTECMOTE en pos de obtener un simulador apropiado para estas redes. El objetivo de proyecto es obtener una herramienta que permita conseguir datos estadísticos importantes de las redes inalámbricas de sensores con nodos CRTECMOTE como conectividad, consumo de potencia y confiabilidad en los datos, previo a su instalación.

Todas las redes implementadas con nodos CRTECMOTE en el simulador poseen una configuración fuente-sumidero en las que la información fluye a un nodo central, coordinador o sumidero. Estas redes no tienen un límite de nodos a utilizar pero si un número máximo de 4 saltos direccionar la información hacia el sumidero.

Los componentes base de un nodo CRTECMOTE son el microcontrolador con arquitectura PIC32MX y un transceptor MRF24J40MA, elementos que debieron modelarse para ser recreados en el simulador. Además, el protocolo de control de acceso al medio (MAC) utilizado en los nodos es el IEEE 802.15.4 en donde un nodo coordinador, en este caso el sumidero, crea una red de área personal (PAN) de bajo consumo a la cual se asocian los demás nodos por medio de una sincronización con faros guía o beacon. Dentro de la red es posible colocar nodos con funciones de creación de redes secundarias para ampliar el área de cobertura de la red e implementar enrutamientos multisalto.

Palabras clave: Castalia, IEEE 802.15.4, nodos CRTECMOTE, Omnet++, red de área personal, transceptores de radiofrecuencia.

## Summary

This document provides a detailed description of the restructuring process carried out at the Castalia simulation tool to adapt it for wireless sensor networks with CRTECMOTE nodes looking for a suitable simulator for this specific kind of network. The project objective is to obtain a tool to get important statistical data of wireless sensor networks with CRTECMOTE nodes as connectivity, power consumption and reliability of the data before the installation of the net.

All networks implemented with CRTECMOTE nodes in the simulator have a source-sink configuration in which the information flows to a central node also call sink coordinator. These networks don't have a maximum number of nodes to be used but they have a limited of 4 hops to route the information to the sink.

The base components of a CRTECMOTE node are the PIC32MX based architecture microcontroller and the MRF24J40MA transceiver, elements that were modeled to be recreated in the simulator. In addition, the protocol used in the nodes for media access control (MAC) was the IEEE 802.15.4 where a coordinator, in this case the sink, created a low consumption personal area network (or PAN) at which other nodes associated to through a beacon-enabled synchronization. Within the network, it can be nodes with secondary networking functions to expand the area of network and implement multi-hop routing.

Keywords: Castalia, IEEE 802.15.4, CRTECMOTE nodes, Omnet++, personal area network, radio frequency transceivers.

## Dedicatoria

*Al ser supremo,  
por todas las bendiciones que me ha otorgado.*

*A mi madre Lidiette,  
quien siempre interpuso nuestro bienestar al propio.  
Mi mejor amiga, consejera y modelo a seguir.*

*A mis hermanos Ariel y Diego,  
quienes me mantuvieron con los pies en la tierra.*

*A mi novio William,  
quien siempre me ayudo a levantarme cuando caía.*

## **Agradecimiento**

A todos los profesores de la escuela de Electrónica, que no solo colaboraron en mi formación académica sino que también me infundieron perseverancia, ansias de aprender y espíritu competitivo.

A mi asesor de proyecto, Ing. Johan Carvajal Godínez, le debo un agradecimiento muy especial pues constituyó un apoyo vital y una guía sin la que este proyecto no hubiera podido ser llevado a cabo.

A todo el personal de SESLab, tanto compañeros estudiantes como profesores, quienes me tendieron una mano durante la realización de este proyecto.

A todas aquellas personas que de una manera u otra colaboraron para que este proyecto saliera adelante.

Infinitas Gracias

# ÍNDICE GENERAL

<b>Capítulo 1</b>	<b>Introducción .....</b>	<b>6</b>
<b>1.1.</b>	<b>Problema existente e importancia de su solución.....</b>	<b>6</b>
<b>1.1.1.</b>	<b>Definición del problema.....</b>	<b>6</b>
<b>1.1.2.</b>	<b>Síntesis del problema .....</b>	<b>7</b>
<b>1.2.</b>	<b>Solución seleccionada .....</b>	<b>8</b>
<b>Capítulo 2</b>	<b>Meta y objetivos .....</b>	<b>10</b>
<b>2.1.</b>	<b>Meta.....</b>	<b>10</b>
<b>2.2.</b>	<b>Objetivo general.....</b>	<b>10</b>
<b>2.3.</b>	<b>Objetivos específicos .....</b>	<b>10</b>
<b>Capítulo 3</b>	<b>Marco teórico.....</b>	<b>11</b>
<b>3.1.</b>	<b>Herramienta de simulación Omnet++.....</b>	<b>11</b>
<b>3.2.</b>	<b>Simulador de redes de sensores Castalia .....</b>	<b>13</b>
<b>3.3.</b>	<b>Protocolo de comunicación inalámbrica IEEE 802.15.4 .....</b>	<b>17</b>
<b>3.4.</b>	<b>Redes inalámbricas de sensores CRTECMOTE.....</b>	<b>20</b>
<b>Capítulo 4</b>	<b>Procedimiento metodológico.....</b>	<b>22</b>
<b>4.1.</b>	<b>Métodos y actividades.....</b>	<b>22</b>
<b>4.1.1.</b>	<b>Análisis de herramienta Castalia y módulos que la componen .....</b>	<b>22</b>
<b>4.1.2.</b>	<b>Análisis de estructura de nodos CRTEcmote .....</b>	<b>22</b>
<b>4.1.3.</b>	<b>Correlación de nodos CRTECMOTE con nodos generados por Castalia .....</b>	<b>23</b>
<b>4.1.4.</b>	<b>Reestructuración de módulos en Castalia.....</b>	<b>23</b>
<b>4.1.5.</b>	<b>Implementación y evaluación de configuración base.....</b>	<b>24</b>
<b>4.1.6.</b>	<b>Definición e implementación de diferentes configuraciones .....</b>	<b>24</b>
<b>4.1.7.</b>	<b>Evaluación de la mejor configuración .....</b>	<b>24</b>
<b>Capítulo 5</b>	<b>Descripción detallada de la solución .....</b>	<b>25</b>
<b>5.1.</b>	<b>Descripción global de cambios en Castalia .....</b>	<b>25</b>
<b>5.2.</b>	<b>Reestructuración de módulos en herramienta Castalia .....</b>	<b>26</b>
<b>5.2.1.</b>	<b>Módulo de Aplicación .....</b>	<b>28</b>
<b>5.2.2.</b>	<b>Módulo de enrutamiento.....</b>	<b>30</b>
<b>5.2.3.</b>	<b>Módulo de Media Access Control (MAC) .....</b>	<b>32</b>

5.2.4. Módulo Radio.....	39
5.2.5. Módulo de manejo de recursos.....	40
5.3. Relación entre módulos dentro de nodos en Castalia .....	42
5.4. Cálculo de pérdidas por distancia en comunicaciones .....	43
5.5. Implementación de configuraciones de prueba y configuración de red base .....	44
5.6. Implementación de configuraciones complementarias .....	46
5.7. Creación de interfaz gráfica.....	48
Capítulo 6 Análisis de Resultados.....	50
6.1. Resultados .....	50
6.1.1. Pruebas de funcionamiento .....	50
6.1.2. Funcionamiento de configuración de red base .....	52
6.1.3. Funcionamiento de redes complementarias .....	54
6.2. Análisis .....	61
6.2.1. Conectividad.....	61
6.2.2. Conexiones multisalto .....	62
6.2.3. Potencia .....	64
6.2.4. Comparación entre redes .....	64
Capítulo 7 Conclusiones y recomendaciones .....	67
7.1. Conclusiones .....	67
7.2. Recomendaciones .....	68
Bibliografía.....	69
Apéndices .....	70
A.1 Glosario.....	70
A.2 Guía de usuario de Castalia .....	71

## INDICE DE FIGURAS

<b>Figura 3.2.1.</b> Modelo de estructura básico de Castalia <sup>[5]</sup> .....	14
<b>Figura 3.2.2.</b> Jerarquía de módulos tipo <i>ned</i> de Omnet++ utilizados por Castalia	15
<b>Figura 3.2.3.</b> Estructura de los nodos de sensado en Castalia <sup>[5]</sup> .....	15
<b>Figura 3.3.1.</b> Estructura simple de superframe para comunicación con trama guía <sup>[7]</sup> .....	18
<b>Figura 3.3.2</b> Subdivisión de periodo activo de un superframe <sup>[7]</sup> .....	19
<b>Figura 3.4.1</b> Arquitectura de red utilizada para probar la tecnología CRTECMOTE .....	20
<b>Figura 5.1.1</b> Correlación de módulos en Castalia con nodos CRTecmote .....	26
<b>Figura 5.2.1</b> Formato de archivos de texto con parámetros de operación de módulo radio en Castalia. Dispositivo CC2420 de TI .....	28
<b>Figura 5.2.2.</b> Diagrama de flujo para el funcionamiento del módulo de aplicación	29
<b>Figura 5.2.3</b> Formato de archivo de texto para parámetros de módulo de aplicación para el microcontrolador PIC32MX46F512L .....	30
<b>Figura 5.2.4</b> Topología de red de árbol de grupos aplicada al proyecto <sup>[7]</sup> .....	33
<b>Figura 5.2.5</b> Configuración básica de árbol de grupos .....	33
<b>Figura 5.2.6</b> Proceso de asociación con tramas guía a PAN principal y PAN secundaria.....	34
<b>Figura 5.2.7</b> Relación entre beacon de entrada (principal) y beacon de salida (secundario) en un nodo FFD <sup>[7]</sup> .....	35
<b>Figura 5.2.8</b> Diagrama de flujo para el proceso de generación de beacon, asociación a PAN y generación de beacon secundario en nodos FFD .....	36
<b>Figura 5.2.9</b> Formato de encabezado de tramas tipo beacon en Castalia.....	37
<b>Figura 5.2.10.</b> Estructura GTSspec .....	38
<b>Figura 5.2.11.</b> Diagrama de flujo del funcionamiento del módulo de manejo de recursos.....	41
<b>Figura 5.5.1.</b> Configuración de prueba para conectividad .....	44
<b>Figura 5.5.2.</b> Configuración de prueba para generación de PAN secundaria .....	45
<b>Figura 5.5.3.</b> Configuración de prueba para multisalto.....	45
<b>Figura 5.6.1.</b> Configuración compleja de red con nodo sumidero en el extremo y una sola PAN secundaria convergente.....	46
<b>Figura 5.6.2.</b> Configuración compleja de red con nodo sumidero en el extremo dos PAN secundarias convergentes .....	46

<b>Figura 5.6.3.</b> Configuración compleja de red con nodo sumidero en el extremo y cuatro PAN secundarias convergentes.....	47
<b>Figura 5.7.1.</b> Interfaz gráfica implementada para simulador de nodos CRTECMOTE.....	48
<b>Figura 5.7.2.</b> Ventana emergente para configuración del protocolo MAC 802.15.4 .....	49
<b>Figura 5.7.3.</b> Ventana emergente para colocación manual de nodos en la red .....	49
<b>Figura 6.1.1.</b> Gráfico de tramas guía recibidos por dispositivos finales para configuración de la figura 5.5.1 con diferentes distancias entre nodos .....	51
<b>Figura 6.1.2.</b> Gráfico de energía promedio de nodos para configuración de red base en Watts.....	53
<b>Figura 6.1.3.</b> Gráfico de energía consumida por los nodos para configuración de red base en Joules .....	53
<b>Figura 6.1.4.</b> Gráfico de energía promedio de nodos para configuración .....	55
<b>Figura 6.1.5.</b> Gráfico de energía consumida por los nodos para configuración de figura 5.6.1 en Joules .....	55
<b>Figura 6.1.6.</b> Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.1 en segundos .....	56
<b>Figura 6.1.7.</b> Gráfico de energía promedio de nodos para configuración .....	57
<b>Figura 6.1.8.</b> Gráfico de energía consumida por los nodos para configuración de figura 5.6.2 en Joules .....	58
<b>Figura 6.1.9.</b> Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.2 en segundos .....	58
<b>Figura 6.1.10.</b> Gráfico de energía promedio de nodos para configuración .....	59
<b>Figura 6.1.11.</b> Gráfico de energía consumida por los nodos para configuración de figura 5.6.3 en Joules .....	60
<b>Figura 6.1.12.</b> Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.3 en segundos .....	60

## ÍNDICE DE TABLAS

<b>Tabla 3.1.1</b> Principales funciones para el manejo de datos en Omnet++ .....	12
<b>Tabla 3.2.1</b> Módulos base creados por desarrolladores de Castalia .....	16
<b>Tabla 3.4.1.</b> Corriente y potencia consumida para un nodo coordinador <sup>□</sup> .....	21
<b>Tabla 3.4.2.</b> Mediciones de potencia para cálculo de enlaces ante variaciones de la distancia de transmisión .....	21
<b>Tabla 5.2.1.</b> Parámetros usados en archivo de parametrización de Castalia para dispositivo MRF24J40.....	39
<b>Tabla 6.1.1.</b> Tramas guía recibidos por dispositivos finales paquetes recibidos por sumidero para configuración de la figura 5.5.1 .....	50
<b>Tabla 6.1.2.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.5.2.....	51
<b>Tabla 6.1.3.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.5.3.....	52
<b>Tabla 6.1.4.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de red base.....	52
<b>Tabla 6.1.5.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.1 .....	54
<b>Tabla 6.1.6.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.2.....	57
<b>Tabla 6.1.7.</b> Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.3.....	59

# Capítulo 1 Introducción

## 1.1. Problema existente e importancia de su solución

### 1.1.1. Definición del problema

El Laboratorio de Sistemas Electrónicos para la Sostenibilidad (SESLab) es un laboratorio de investigación perteneciente a la Escuela de Electrónica del Instituto Tecnológico de Costa Rica en donde profesionales de diferentes áreas, entre las que se pueden mencionar Electrónica y Diseño Industrial, buscan adaptar procesos y desarrollar sistemas electrónicos para fomentar la sostenibilidad, todo esto para su utilización en la conservación ambiental.

Uno de los proyectos de este laboratorio que se ha venido desarrollando desde el 2009 consiste en el diseño e implementación de redes inalámbricas de sensores (WSN por sus siglas en inglés), utilizando herramientas de uso abierto o “libres”, que permitan tanto la monitorización de variables ambientales como un posible uso en monitoreo de infraestructuras.

Desde la concepción del proyecto, llamado “Desarrollo tecnológico de un sistema de adquisición de datos ambientales para su uso en proyectos de investigación científica: Arquitectura abierta CRTECMOTE”, se ha buscado generar un sistema capaz de realizar mediciones confiables e inteligentes y que a su vez mantenga un funcionamiento de bajo consumo, presente una inversión de bajo costo y características escalables [1].

Para el desarrollo de este proyecto, por su complejidad y alcance, se ha dividido el trabajo en diferentes subproyectos que han venido a representar avances semestrales realizados en conjunto por estudiantes y profesores para el desarrollo global del sistema, ya sea en concepto de asistencias estudiantiles o como proyectos finales de graduación de la escuela de Electrónica.

Actualmente, se tiene un prototipo de red inalámbrica de sensores utilizando sensores CRTECMOTE en una topología tipo estrella. Esta implementación cuenta con tres nodos CRTECMOTE y un nodo sumidero o LOADPOINT utilizado para el control del flujo de la información dentro de la red. Este nodo sumidero se comunica por medio del protocolo ETHERNET a una computadora portátil (PC) desde donde se puede acceder a los datos de manera remota.

Debido a su alcance, la tecnología de redes inalámbricas de sensores CRTECMOTE ha sido incorporada recientemente en el proyecto “Predicción remota de fallas estructurales en puentes: eBRIDGE”, como base de la red de monitoreo que se implementará para la recolección y enrutamiento de los datos.

Como se mencionó anteriormente, dentro de los avances del proyecto se dio la implementación prototipo de una WSN con topología tipo estrella. Sin embargo, no se enfatizó en la certeza de los datos sensados y en la confiabilidad que se puede tener en los procesos de comunicación para redes de sensores inalámbricas utilizando nodos CRTECMOTE. Es necesario conocer los datos de eficiencia de la red para determinar los alcances y limitaciones del proyecto global.

### **1.1.2. Síntesis del problema**

No se tiene conocimiento del comportamiento, en términos de eficiencia, localización de dispositivos, área de alcance de estos nodos, consumo de potencia, tiempo de muestreo y frecuencia de operación; de las redes inalámbricas de sensores con nodos CRTECMOTE para aplicaciones de monitorización ambiental y de infraestructura.

## 1.2. Solución seleccionada

Existen dos maneras básicas de conocer el comportamiento de un sistema: mediante las mediciones de los parámetros del mismo luego de su implementación, o por medio de simulaciones previas a su instalación.

Para la gran mayoría de los casos, el instalar un sistema antes de alguna prueba que corrobore su correcto funcionamiento es altamente costoso y no necesariamente generarán los resultados esperados. En el caso particular para las redes de sensores inalámbricas se corre el riesgo de encontrar problemas de transmisión en los datos, desacople de nodos de sensores o incluso el fallo completo de la red; esto por no mencionar el costo innecesario que representaría reparar fallas que se habrían podido evitar mediante una simulación de la red antes de su implementación. Es por esta razón que la simulación se postula como la opción más acertada pues da solución a todos los problemas anteriormente mencionados al tener la posibilidad de recrear una red desde un sistema computacional, todo esto con un costo monetario muy bajo.

Se debe tener presente que para el diseño del simulador propuesto existen algunos requerimientos que se deben cumplir:

- Implementar el protocolo IEEE 802.15.4 para la capa MAC en la banda de 2.4GHz.
- Habilitar las funciones principales del receptor/transmisor MRF24J40
  - Identificación de dispositivos
  - Re-sincronización de dispositivos
- Enrutamiento multi-salto de información a través de la red.
- Generación de redes secundarias para abarcar zonas más allá del área de cobertura de los dispositivos.
- Emular el consumo de energía del módulo total en base a los dispositivos que conforman el CRTECMOTE (PIC32MX460F512L y MRF24J40).

Debido a la función que tendrá el simulador, las limitaciones del mismo estarán ligadas a los dispositivos que se utilizarán en la red que se desee simular y la región geográfica que abarcará la misma. En base a esta premisa se tienen las siguientes limitaciones:

- El sistema deberá ser reproducible.
- Escalabilidad
- Posibilidad de seleccionar otros dispositivos para conformar los nodos CRTECMOTE (en lugar del PIC32MX460F512L y/o MRF24J40).

En el desarrollo de la solución se buscan implementar diferentes configuraciones de redes inalámbricas de sensores por medio de la herramienta de simulación escogida para determinar la eficiencia energética y conectividad de una red determinada antes de que la misma sea instalada.

Cabe destacar que, aunque el desarrollo de una configuración que relacione varios nodos CRTECMOTE y la determinación su funcionamiento por medio de simulaciones es la solución buscada, el modelado del comportamiento de un nodo de sensores CRTECMOTE es la base de este proyecto.

Se debe enfatizar que aunque los nodos CRTECMOTE utilizan el estándar IEEE 802.15.4 para el acceso al medio, no tiene implementaciones multi-salto desarrolladas. Sin embargo, es uno de los procesos que se busca implementar en el futuro por lo que deberá ser tomado en cuenta en la solución.

## **Capítulo 2 Meta y objetivos**

### **2.1. Meta**

Modelar componentes de simulación que permitan definir el comportamiento de una red inalámbrica de sensores para aplicaciones de monitorización estructural y/o ambiental.

### **2.2. Objetivo general**

Desarrollar un modelo referencial de redes inalámbricas de sensores que permita comparar el comportamiento de diferentes redes CRTECMOTE por medio de simulaciones en la herramienta Castalia.

### **2.3. Objetivos específicos**

- Desarrollar un modelo de simulación del nodo CRTECMOTE, compatible con la herramienta Castalia, para su utilización en configuraciones de redes inalámbricas de sensores.
- Establecer un esquema de referencia de simulación, para la herramienta Castalia, de una red inalámbrica de sensores con nodos CRTECMOTE
- Determinar la mejor configuración de red inalámbrica de sensores con nodos CRTECMOTE por medio de simulación del modelo con diferentes parametrizaciones.

## Capítulo 3 Marco teórico

Diferentes herramientas se han desarrollado para la simulación de redes inalámbricas. Como criterios de selección de dicha herramienta se tuvieron la necesidad de conocer el código fuente de la misma (código abierto), la orientación a módulos (de ser posible a capas según el estándar OSI) y la confiabilidad de los resultados en las simulaciones.

### 3.1. Herramienta de simulación Omnet++

La herramienta Omnet++ consiste en un ambiente de simulación de eventos de tipo discreto que se ha desarrollado con un énfasis en redes de comunicación, tanto cableadas como inalámbricas. Aunque originalmente no fue concebido como un simulador de redes propiamente posee una estructura modular orientada al uso de capas y funciones altamente versátiles de empaquetado, desempaquetado y comunicación intermodular de datos lo cual lo coloca como una herramienta confiable y potente de simulaciones de redes, la cual además puede ser reestructurada de manera sencilla por un programador con dominio básico del lenguaje C++ Además, este software no requiere de licencia adquirida (software libre). Estas características hacen que el software de simulación de eventos OMNET++ se presente como la opción más acertada para el desarrollo del proyecto pues se ajusta a las características de tecnología abierta, escalabilidad y compatibilidad del proyecto global.

Omnet++ basa su funcionamiento en la creación de archivos con extensión .ned (propios de Omnet++) los cuales representan una entidad o dispositivo, el cual se definirá como módulo a partir de este punto. Un archivo o módulo ned puede representar por ejemplo un nodo en la red, una capa de comunicación dentro del nodo o incluso la red global dentro de la cual el dispositivo se encuentra. Dentro de este archivo se definen las salidas y entradas del módulo por medio de los cuales se interconectará con otros módulos, con submódulos dentro

del mismo o con módulos superiores. A cada uno de estos archivos .ned se les asigna un código fuente en lenguaje C++ (tanto el encabezado con extensión .h como su fuente en extensión .cc) el cual contendrá todas las funciones que el módulo podrá desempeñar.

Otra característica importante de la herramienta Omnet++ es la versatilidad en la comunicación de paquetes entre sus módulos. Esto se logra por medio de la creación de archivos de extensión *msg* (al igual que los archivos .ned, propios de Omnet++) que representan las tramas de información para la comunicación entre nodos, conformados por encabezados de trama y por la información que se desea transmitir. Estos archivos con extensión .msg son utilizados en las clases del código C++ para la creación de objetos que representen las tramas de información que se comunicarán entre módulos. Se debe enfatizar que las tramas no se transmiten propiamente entre módulos como si se tratará de una carta, sino que se duplican de módulo a módulo.

De igual manera que los archivos de extensión ned, todo archivo de extensión msg o de mensaje posee su propio código fuente en C++ por medio del cual se logra empaquetar y desempaquetar los datos, definir o obtener un espacio determinado dentro de la trama (destinatario o fuente del mensaje por ejemplo) o cambiar la longitud de los encabezados.

Se deben destacar de Omnet++, además, sus funciones para el manejo de información entre módulos. En la tabla 3.1.1 se detallan las funciones de Omnet++ que se deben conocer para comprender este proyecto.

**Tabla 3.1.1** Principales funciones para el manejo de datos en Omnet++

<b>Función</b>	<b>Descripción</b>
Send (pkt, name)	Envía mensajes a otros módulos
sendDirect (pkt, name)	Envía mensaje a otros módulos ignorando conexiones modulares
encapsulatePacket (pkt&header, pkt)	Acopla encabezado de módulo al inicio de los datos
decapsulatePacket (pkt)	Elimina el encabezado acoplado por el módulo y recupera datos

Una vez que se hayan definido los módulos de Omnet++ y los mensajes que se utilizarán para comunicarlos es necesario determinar los parámetros de inicialización de la red. Esto se logra por medio de la implementación de archivos de inicialización llamados *omnetpp.ini* en los cuales se especifican estos parámetros (tiempo de simulación, número de mensajes, entre otros). Se debe enfatizar que pueden existir varias configuraciones dentro de un solo archivo de inicialización, con lo que se pueden recrear varios ambientes de simulación en una sola ejecución. Sin embargo, sin importar el número de configuraciones siempre debe existir la configuración *General* que funcionará por defecto.

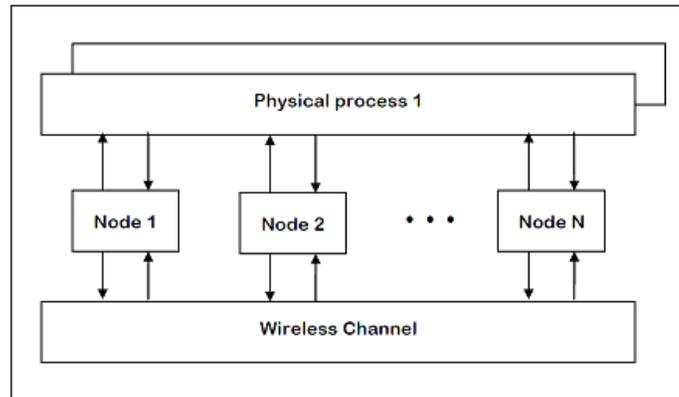
Ahora bien, la herramienta OMNET++ por sí sola no se ajusta a las necesidades del proyecto, esto por la cantidad y la densidad de información en las redes de sensores, el cambio abrupto en los datos, el bajo consumo de potencia y la distribución y configuración que presentan los nodos que componen a la misma. En proyectos de investigación pasados, otras entidades educativas han dado solución a este problema creando la herramienta de simulación Castalia [5] como complemento de OMNET++.

### **3.2. Simulador de redes de sensores Castalia**

Castalia es un simulador específicamente para redes inalámbricas de sensores, redes de área corporal (usadas en medicina) y cualquier otra red afín en la que se utilice un consumo bajo de potencia. Está basada en OMNET++ y puede considerarse como una extensión de este el cual, al igual que su software base, es de código abierto.

El funcionamiento global de Castalia comprende la formación de nodos que representan los dispositivos de sensado dentro de la red, según se puede apreciar en la figura 3.2.1. Además, se generan los módulos wireless channel y physical process que emulan el medio de propagación de las señales (aire generalmente) y

los procesos físicos dentro del mismo como el paso de un objeto a través de la red, respectivamente.

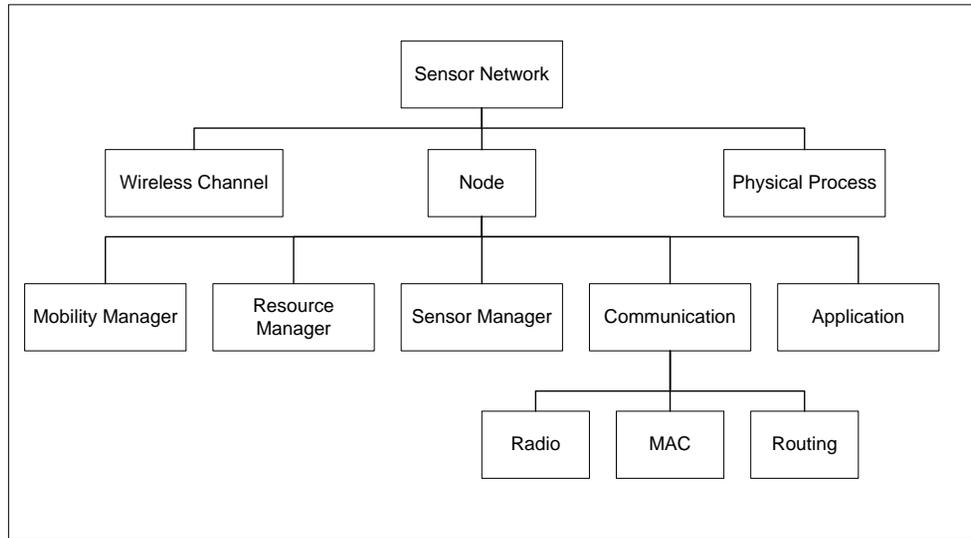


**Figura 3.2.1.** Modelo de estructura básica de Castalia <sup>[5]</sup>

Cada uno de los bloques dentro de la figura 3.2.1 representan un módulo (archivo de extensión .ned) de Omnet++ y cuyo código fuente fue generado por los desarrolladores de Castalia para crear un simulador orientado a redes de sensado.

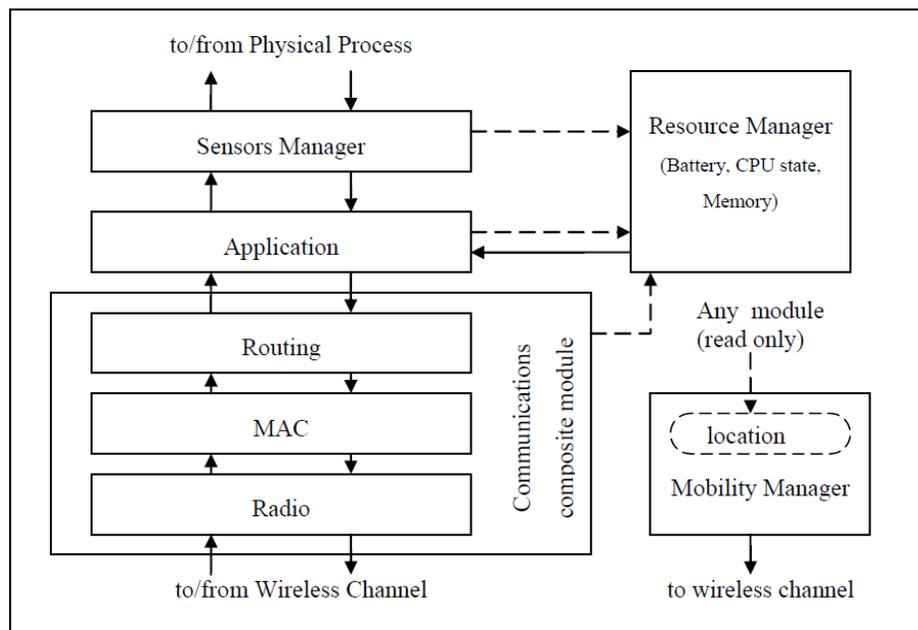
Se puede resumir entonces que Castalia hace uso de las interconexiones de Omnet++ (por medio de la generación de archivos con extensión .ned) para generar redes inalámbricas, de bajo consumo en su mayoría, configurando un número determinado de nodos y su distribución espacial y temporal para luego interconectarlos utilizando los módulos wireless y physical, según las especificaciones del usuario. Castalia, además, compone el código fuente incorporado a cada módulo que regulará dicha el funcionamiento de la red.

La jerarquía de los archivos tipo *ned* se presenta en la figura 3.2.2. Cada uno de estos módulos posee su propio código fuente que determina su funcionamiento. Nótese que Castalia creará un número de módulos *node* (tipo *ned*) igual al número de dispositivos de sensado especificados por el usuario.



**Figura 3.2.2.** Jerarquía de módulos tipo *ned* de Omnet++ utilizados por Castalia

Para adentrarse en la estructura de Castalia se debe comprender el funcionamiento de los módulos que emularán los dispositivos o nodos que componen la red. Para esto se deben conocer los módulos dentro de cada nodo, los submódulos dentro de estos si los hubiera y las interconexiones entre estos y con el exterior. Dentro de cada nodo se tiene la estructura que se presenta en la figura 3.2.3.



**Figura 3.2.3.** Estructura de los nodos de sensado en Castalia <sup>[5]</sup>

El módulo de comunicación comprende las tres funciones básicas de la comunicación inalámbrica: el enrutamiento, el control de acceso al medio y el manejo físico del canal; los cuales son manejados por los módulos de routing, MAC y radio, respectivamente. El módulo de movilidad maneja la posición del nodo en ambientes en donde estos dispositivos se encuentren en movimiento. Las tareas que el nodo realizará se incorporarán por medio del módulo de aplicación y, si la tarea lo necesita, por el módulo de manejo de sensores. Finalmente, el módulo *resource manager* como su nombre lo indica se encarga de calcular la energía que el nodo ha consumido en su operación y la capacidad de memoria con que este cuenta para sus funciones.

Todos los módulos y submódulos dentro de cada nodo pueden ser cambiados a conveniencia al realizar la simulación por medio de los archivos de configuración *omnetpp.ini*. De esta manera, con la misma herramienta es posible simular nodos de sensado cuya aplicación sea sensar un convertidor analógico digital o enviar paquetes de manera periódica cambiando el módulo de aplicación de la figura 3.2.2, por ejemplo.

Actualmente existen aplicaciones y submódulos de comunicación de MAC y enrutamiento creados por los desarrolladores de Castalia como base para futuros proyectos. Estos se resumen en la tabla 3.2.1.

**Tabla 3.2.1** Módulos base creados por desarrolladores de Castalia <sup>1</sup>

<b>Application</b>	<b>Routing</b>	<b>MAC</b>
BridgeTest	BypassRouting	BypassMAC
conectivityMap	MultiPathRouting	TMAC
simpleAggregation		tunableMAC
throughputTest		Baseline MAC
valuePropagation		
valueReporting		

<sup>1</sup> Para mayor detalle de cada uno de estos módulos consultar referencia [5].

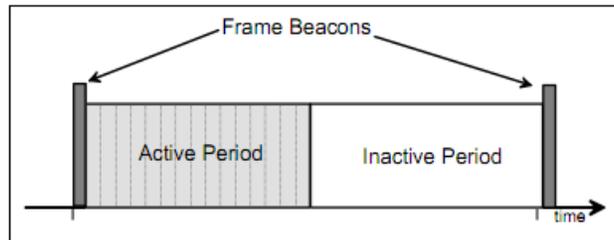
### **3.3. Protocolo de comunicación inalámbrica IEEE 802.15.4**

El estándar IEEE 802.15.4 fue desarrollado para redes de tipo WPAN o de área personal que mantengan un bajo consumo de potencia y cuyas transmisiones sean de baja velocidad (entre los 20 y 250Kbps). Debido a estas características, este estándar es altamente utilizado en redes de sensores de carácter inalámbrico. Sirve como protocolo de comunicación para la capas MAC y PHY según el modelo de capas para sistemas abiertos u OSI.

Dentro de las especificaciones del estándar la IEEE definió dos tipos básicos de dispositivos para componer la red: dispositivos de función o funcionalidad completa o FFD (por sus siglas en inglés) y los dispositivos de función reducida o RFD. Los FFD poseen la posibilidad de actuar de tres maneras diferentes: como coordinadores principales de la red de área personal, llamadas PAN; como coordinadores de una PAN secundaria o como simples dispositivos finales. Los RFD, por el otro lado, solo pueden funcionar como dispositivos finales. Utilizando esta caracterización, el protocolo es capaz de soportar topologías en estrella y punto a punto, las cuales pueden ser extendidas a topologías de árbol de grupos (cluster tree) o mallas.

Para la creación de redes de área personal, el protocolo especifica la asociación de dispositivos por medio de dos opciones: por medio de la generación de tramas guía también llamados beacons o sin estos. Un trama guía o beacon constituye una forma de sincronización para las redes WPAN. Están formados por una trama cuyas especificaciones en el encabezado la definan como beacon. En las redes funcionando con tramas guía, los dispositivos FFD que se hayan definido como coordinadores principales de la red generarán una trama de tipo beacon y la transmitirán periódicamente a todos los dispositivos a su alcance mediante una transmisión broadcast sin necesidad de mecanismos de control de acceso al medio, mientras que el resto de FFD tienen la posibilidad de generar beacon secundarios sincronizados con el beacon principal, si así se requiriera. Cada

beacon representa el inicio de un nuevo super encapsulado o marco de tiempo de transmisión, como se puede observar en la figura 3.3.1.



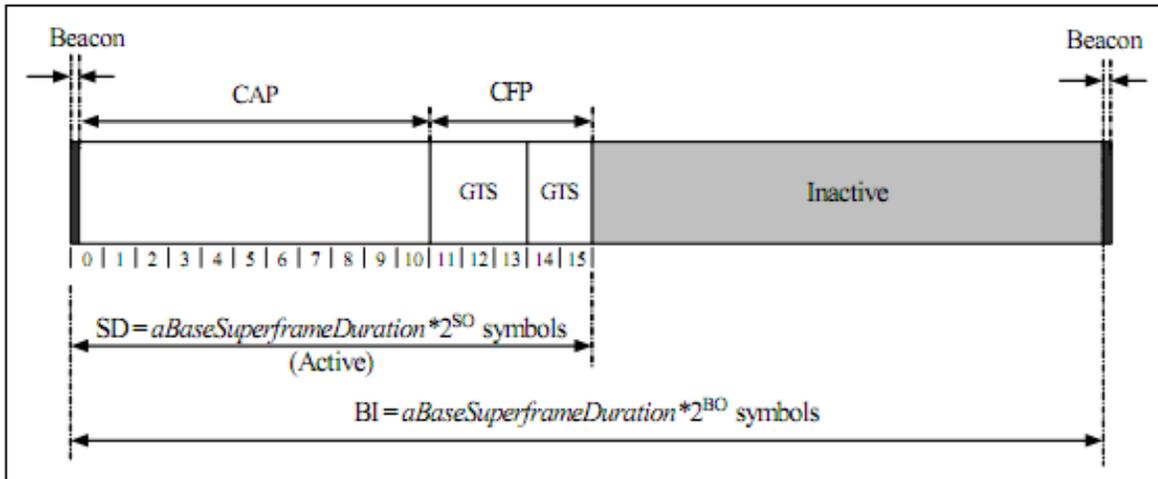
**Figura 3.3.1.** Estructura simple de super encapsulado para comunicación con trama guía <sup>[7]</sup>

En la figura 3.3.1 se denotan además, los dos periodos básicos en los que se dividen los super encapsulados: activo e inactivo. Estos periodos representan los espacios de tiempo en el que los dispositivos pueden transmitir datos en la red o deben permanecer en reposo, respectivamente. En cada parte activa del super encapsulado se tienen 16 espacios, llamados comúnmente ranuras (*slots*), cada uno conformado por un número de símbolos de comunicación determinados por el estándar. Para definir la duración de los periodos entre tramas guía y para el periodo activo del super encapsulado se definen 4 parámetros en el estándar IEEE: *aNumSuperframeSlots* que representa el número de slots o espacios en el periodo activo, *aBaseSlotDuration* que constituye el número de símbolos dentro de cada slot y los parámetros *beaconOrder* y *frameOrder* los cuales determinan los intervalos entre beacon y el periodo activo, respectivamente. A partir de estos se obtienen los parámetros *beaconInterval* y *frameInterval* que definen los intervalos según las ecuaciones 3.3.1 y 3.3.2. Al multiplicar estos intervalos con el tiempo de duración (en segundos) se obtiene el tiempo de cada periodo.

$$frameInterval = aNumSuperframeSlots * aBaseSlotDuration * 2^{frameOrder} \quad (3.3.1)$$

$$beaconInterval = aNumSuperframeSlots * aBaseSlotDuration * 2^{beaconOrder} \quad (3.3.2)$$

Ya se ha definido que el `frameInterval` junto con el tiempo de cada símbolo define el tiempo de duración del periodo activo de los super encapsulados. Ahora bien, la parte activa del super encapsulado se subdivide según lo muestra la figura 3.3.2.



**Figura 3.3.2** Subdivisión de periodo activo de un super encapsulado [7]

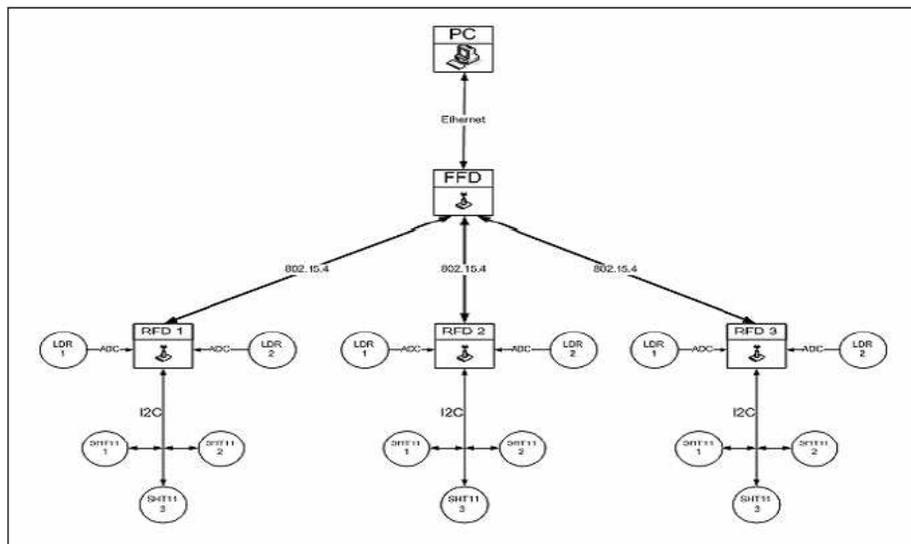
El periodo de contención de acceso o CAP, por sus siglas en inglés, inicia inmediatamente después del beacon como se observa en la figura 3.2.2. Si no se tuvieran espacios de tiempo garantizados o GTS abarcaría todo el periodo activo del super encapsulado. Todo mensaje que se desee transmitir en el CAP deberá ser incorporado al canal por medio del algoritmo CSMA-CA, a menos que se trate de un mensaje de reconocimiento (acknowledge).

En caso que se tengan espacios de tiempo garantizados o GTS por sus siglas en inglés, estos corresponderán a periodo de libre de contención o CFP. Para agregar este periodo no se debe tomar tiempo de la parte inactiva del super encapsulado sino de la activa, en un caso inicial del CAP. Para la transmisión de datos en este periodo no hay necesidad de utilizar el CSMA-CA.

### 3.4. Redes inalámbricas de sensores CRTECMOTE

Para la validación de los datos obtenidos por medio de las simulaciones se necesita comparar éstos con mediciones realizadas al nodo real CRTECMOTE y comprobar que la configuración dada por el modelo realizado en Castalia. Las mediciones del nodo real se realizaron en proyectos anteriores que han sido parte del proyecto global (véase referencias [1], [2] y [3]).

Como base de todas las configuraciones que se desarrollarán se tiene la topología estructurada en estrella con un nodo sumidero que fue implementada al darle aplicaciones Miwi a los nodos CRTECMOTE [3], como la que se presenta en la figura 3.4.1.



**Figura 3.4.1** Arquitectura de red utilizada para la tecnología CRTECMOTE [1]

Gracias a esta implementación ya se cuenta con valores de comportamiento de una red inalámbrica de sensores que sirven de referencia para comparar las redes que se proyectan simular. Resultados importantes que se tomaron en proyectos anteriores y que serán de utilidad en este informe se presentan en las tablas 3.4.1 y 3.4.2.

**Tabla 3.4.1.** Corriente y potencia consumida para un nodo coordinador <sup>[2]</sup>

Corriente promedio (mA)	Potencia (W)
0.08264588	0.27273141

**Tabla 3.4.2.** Mediciones de potencia para cálculo de enlaces ante variaciones de la distancia de transmisión

Distancia	Potencia (dBm)	Intentos
0	-40	1
10	-63	1
20	-68	1
30	-76	1
40	-87	1
50	-87	1
60	-88	1
70	-88	1
80	<-90	1
90	<-90	1
100	<-90	1
110	<-90	2
120	<-90	3
130	<-90	3
140	<-90	3
150	<-90	4

A partir de estas redes se progresará hacia otras distribuciones de redes inalámbricas buscando las mejores configuraciones en términos de consumo de potencia y cantidad de saltos entre un nodo y el nodo sumidero de la red.

---

<sup>2</sup> Tablas parciales tomadas de [3].

## **Capítulo 4 Procedimiento metodológico**

### **4.1. Métodos y actividades**

#### **4.1.1. Análisis de herramienta Castalia y módulos que la componen**

El código fuente para el simulador Castalia fue generado hace poco tiempo, relativamente (última revisión realizada en el 2011). Es por esta razón que antes de cualquier diseño o implementación se debe conocer en detalle el funcionamiento de los módulos de Castalia para así poder vincular los requerimientos y limitaciones del proyecto con lo que puede ser realizado por medio de Castalia.

Este análisis se inicio con la lectura del manual de usuario [5], en el cual mencionan de manera superficial las funciones de los diferentes módulos y su relación con otros módulos o submódulos. Además, se detallan los principales comandos para la ejecución de archivos de configuración o “omnetpp.ini” los cuales son usados por Castalia para generar cada uno de los nodos con sus diferentes funciones dentro de la red.

Seguidamente, se realiza una lectura exhaustiva de los archivos de código fuente C++ que componen Castalia para entender su funcionamiento y a la vez documentar dicho código.

#### **4.1.2. Análisis de estructura de nodos CRTecmote**

Posteriormente, se inicia el análisis de la estructura interna de los nodos CRTecmote, tanto en hardware como en software. Dentro de este análisis se deben mencionar la obtención de información de retardos, potencia y funcionamiento general de las hojas de datos del microcontrolador

PIC32MX460F512L y el transmisor/receptor MRF24J40 los cuales componen los nodos CRTECMOTE implementados hasta el momento.

Como parte de la sección de software de los nodos CRTECMOTE se debió analizar el sistema operativo SIWA-RTOS, el cual fue desarrollado como parte del proyecto global en busca de un sistema operativo de bajo consumo [2]. Cabe destacarse que se debió asegurar que las funciones y módulos implementados en el simulador pudieran ser reproducidos en implementaciones reales por lo que las implicaciones de utilizar el sistema operativo SIWA-RTOS debieron ser tomadas en consideración. Sin embargo, todas las implementaciones dentro del simulador fueron tomadas en niveles lógicos superiores por lo que no se profundizó en los estados de operación de las tareas, el tick u otras características de este sistema operativo de manera tan profunda.

#### **4.1.3. Correlación de nodos CRTECMOTE con nodos generados por Castalia**

Una vez analizadas ambas bases del simulador se procedió a hacer la correlación de las mismas para determinar la mejor manera de acoplar todos los requerimientos al simulador Castalia de manera que este emule fielmente los nodos CRTECMOTE.

#### **4.1.4. Reestructuración de módulos en Castalia**

Luego de relacionar los nodos CRTECMOTE reales con los nodos generados en Castalia, se logró definir cuales de los módulos y submódulos dentro de los nodos en Castalia deben ser reestructurados de manera que se emule el comportamiento de los nodos CRTECMOTE tanto en funcionamiento como en términos de consumo de potencia.

#### **4.1.5. Implementación y evaluación de configuración base**

Una vez que se tuvo en Castalia un nodo que recree los nodos reales CRTECMOTE se procedió a implementar la configuración base presentada en la figura 3.4.1, definida en proyectos anteriores. Al implementar esta configuración se desea comprobar que el funcionamiento de la red simulada sea análogo a la red real que se ha puesto en funcionamiento.

#### **4.1.6. Definición e implementación de diferentes configuraciones**

Seguidamente, una vez que se cuente con el modelo del nodo CRTECMOTE adaptado para Castalia se puede proceder a diseñar, comparar y corroborar el funcionamiento de diferentes configuraciones de redes inalámbricas de sensores. Las configuraciones que se implementaron para evaluación fueron definidas en conjunto con el asesor del proyecto Johan Carvajal, buscando posibles configuraciones que optimizarán el funcionamiento de la red.

#### **4.1.7. Evaluación de la mejor configuración**

La herramienta Castalia posee la opción de simular múltiples configuraciones en una sola ejecución lo que permite comparar diferentes distribuciones de red tanto en términos de localización como en funcionamiento o parametrización.

Basándose en términos de consumo de potencia, potencia media y capacidad de conectividad se analizaron diferentes configuraciones buscando la más apropiada para ser implementada en un área determinada y con un número de nodos de sensado fijo.

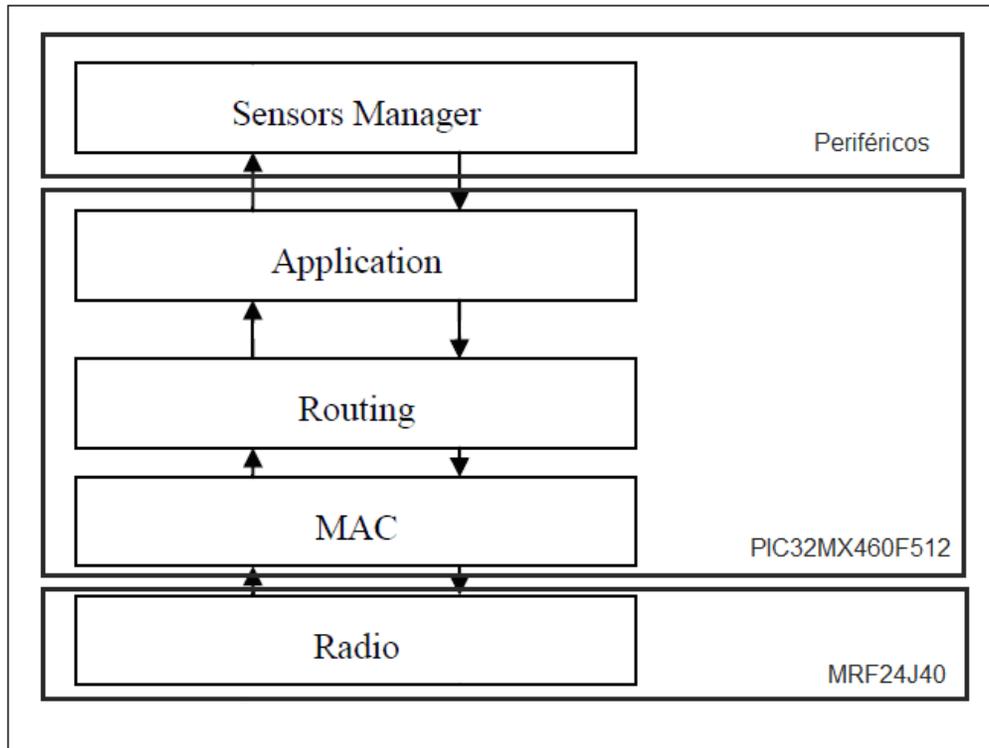
## Capítulo 5 Descripción detallada de la solución

### 5.1. Descripción global de cambios en Castalia

Por medio del análisis de la herramienta Castalia se logró concretar que el funcionamiento básico de este simulador consiste en la interrelación de módulos como se observó en la figura 3.2.2. Utilizando código fuente se definen las funciones de los módulos. Estas funciones pueden variar según la clase que se escoja (véase tabla 3.2.1). Cabe destacar además que existen otras clases auxiliares, utilizadas por cada uno de los bloques de la figura 3.2.2, independientemente del módulo de función que se pretende emular. Estas clases se mencionan a continuación:

- VirtualApplication
- ApplicationPacket\_m
- VirtualMac
- MacPacket\_m
- RadioControlMessage\_m
- RadioSupportFuntions
- VirtualRouting
- RoutingPacket\_m
- VirtualMobilityManager
- MobilityManagerMessage\_m
- ResourceManagerMessage
- SensorManagerMessage

Así, todos los módulos desarrollados para emular la capa MAC (sean el TMAC, SMAC, MAC80154 o cualquier otro desarrollado) utilizan las funciones de la clase auxiliar *VirtualMac.cc* para trabajar correctamente. Esto se aplica equivalentemente para los demás módulos. Al relacionar los módulos de Castalia con los nodos CRTECMOTEse encontró la relación que se resume de manera sencilla en la figura 5.1.1.



**Figura 5.1.1** Correlación de módulos en Castalia con nodos CRTECMOTE

En base a lo observado en la figura 5.1.1 se debió lograr que el módulo de radio de Castalia emule el funcionamiento de los transmisores/receptores y los módulos de MAC, enrutamiento y aplicación de Castalia constituyeran lo aportado por el microcontrolador, todo esto tanto en aspectos de funcionamiento como en consumo de potencia.

## 5.2. Reestructuración de módulos en herramienta Castalia

Uniéndolo a los requerimientos y limitaciones del proyecto (véase sección 1.2) se definieron los módulos de Castalia a los que se les debería realizar cambios pequeños o una reestructuración total. Los módulos que debieron reestructurarse para acoplarse con las funciones de un nodo CRTECMOTE fueron:

- MAC
- Routing
- Resource Manager

Además, para la implementación de las tareas básicas de sensado se creó una nueva aplicación o módulo para reproducir el funcionamiento de los nodos CRTECMOTE. Los demás módulos no mencionados no poseen cambios significativos.

Cabe destacar que la solución se orientó a emular los nodos CRTECMOTE pero tratando de mantener el mismo esquema de programación que se ha desarrollado en Castalia de manera que dichos cambios no afecten las demás aplicaciones. En el caso de los parámetros de consumo de potencia, los desarrolladores de Castalia optaron por leer archivos de parametrización (de extensión txt) en los cuales se detallan los parámetros de los dispositivos con lo que se logra una versatilidad en la escogencia de dispositivos.

Por ejemplo, para calcular el consumo de potencia de los transmisores/receptores, los cuales son emulados por Castalia por el submódulo Radio, se realiza la lectura de un archivo de texto con formato definido que contenga el nombre de modo de consumo del dispositivo, su tipo de modulación, velocidad de datos, número de bits por segundo, el ancho de banda, el ruido de banda, la sensibilidad del dispositivo y el consumo de potencia tanto en envío como en recepción de datos del mismo en este modo y los tiempos de atraso al pasar entre modos. En la figura 5.2.1 se muestra el formato que deben tener los archivos de texto para los dispositivos de transmisión/recepción.

Este mismo principio de lectura de archivos de texto que contengan los parámetros de operación de los dispositivos se reprodujo para el manejo de potencia del microcontrolador (módulos de Aplicación, enrutamiento y MAC para Castalia) como se verá más adelante.

```

RX MODES
# Name, dataRate(kbps), modulationType, bitsPerSymbol, bandwidth(MHz),
noiseBandwidth(MHz), noiseFloor(dBm), sensitivity(dBm), powerConsumed(mW)
normal, 250, PSK, 4, 20, 194, -100, -95, 62
IDEAL, 250, IDEAL, 4, 20, 194, -100, -95, 62

TX LEVELS
Tx_dBm 0 -1 -3 -5 -7 -10 -15 -25
Tx_mW 57.42 55.18 50.69 46.2 42.24 36.3 32.67 29.04

DELAY TRANSITION MATRIX
# State switching times (time to switch from column state to row state, in msec)
#      RX      TX      SLEEP
RX     -      0.01  0.194
TX     0.01  -      0.194
SLEEP 0.05  0.05  -

POWER TRANSITION MATRIX
#      RX      TX      SLEEP
RX     -      62      62
TX     62     -      62
SLEEP 1.4    1.4    -

SLEEP LEVELS
idle 1.4, -, -, -, -

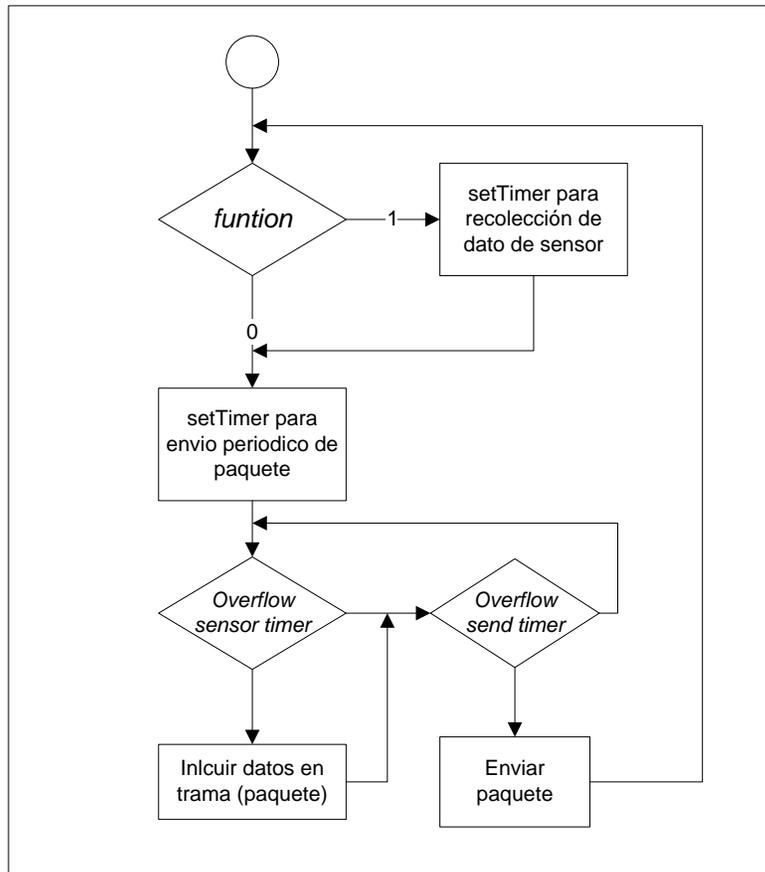
```

**Figura 5.2.1** Formato de archivos de texto con parámetros de operación de módulo radio en Castalia. Dispositivo CC2420 de TI

**5.2.1. Módulo de Aplicación**

Los desarrolladores de Castalia ya han creado módulos de aplicación que realizan diferentes tareas pero ninguno de ellos se ajusta a las tareas realizadas por los nodos CRTECMOTE. Es por esta razón que se debe crear un módulo de aplicación que emule las tareas del nodo CRTECMOTE en Castalia.

Se deseaba comprobar el funcionamiento de los nodos tanto en conectividad (capacidad de los nodos en comunicarse entre ellos, sin importar el contenido de los mensajes) como en confiabilidad de los mensajes. Para esto se adaptó un parámetro de configuración inicial en el módulo de aplicación llamado *function* que se encarga de indicarle al nodo si debe enviar periódicamente tramas sin un contenido definido (pruebas de conectividad) o si debe incorporar a estas tramas datos obtenidos por el módulo de sensores. En la figura 5.2.2 se muestra el diagrama de flujo seguido por el módulo de aplicación implementado.



**Figura 5.2.2.** Diagrama de flujo para el funcionamiento del módulo de aplicación

Para el manejo de potencia de este módulo se tomo como base lo mostrado en la figura 5.1.1, en donde el módulo de aplicación forma parte del consumo de energía que tendría el microcontrolador en los nodos CRTECMOTE reales. Buscando que la solución se adaptara al modo de programación de Castalia se utilizó la lectura desde un archivo de texto de los parámetros del dispositivo que en la realidad contendría al módulo de aplicación de manera que se pueda cambiar de forma sencilla el componente (el microcontrolador en este caso) con el que se desea simular el funcionamiento del nodo. De esta forma, se puede variar el microcontrolador que realizaría las funciones de aplicación del nodo sin alterar el código en Castalia. El formato que deberán tener los archivos de texto para parámetros del módulo de aplicación (además de enrutamiento y MAC como se verá más adelante) se muestra en la figura 5.2.3.

```
# Name, OperationFrequency(MHz), powerConsumed(mW)
Normal, 80, 270
Idle, 80, 115
```

**Figura 5.2.3** Formato de archivo de texto para parámetros de módulo de aplicación para el microcontrolador PIC32MX46F512L

Como se observa en el formato de la figura 5.2.3, es posible tener varios modos de consumo de energía. En este caso se tomaron los modos de consumo normal de energía y el modo de consumo *idle* del microcontrolador (véanse referencias [2] y [10]) para estados de reposo con los que se busca emular el ahorro energético de los nodos CRTECMOTE implementado en proyectos anteriores [2]. Esta misma caracterización puede ser implementada con otras frecuencias de operación del microcontrolador, siempre y cuando los datos incorporados en el archivo de parámetros sean congruentes con lo estipulado por el fabricante del dispositivo.

Para buscar el correcto funcionamiento del módulo en términos de manejo de potencia se utilizó la misma premisa que en los nodos CRTECMOTE reales en donde al realizar alguna tarea el dispositivo se encuentra en el modo de consumo normal y al no tener ninguna función que realizar (en reposo) el modo cambia a *idle*.

## 5.2.2. Módulo de enrutamiento

Los módulos de enrutamiento poseen las funciones, como su nombre lo indica, de encontrar el camino adecuado para los datos y enrutar los mismos dentro de la red. Generalmente usan los parámetros de dirección de destino y de fuente de los encabezados de las tramas de este módulo (definidos en los archivos de tipo mensaje o msg de Castalia) para determinar el próximo nodo al que se deberán direccionar los datos.

Los módulos diseñados actualmente en Castalia dejan todas las funciones de enrutamiento al módulo de routing y no delega ninguna de estas a las capas de radio o MAC. O sea que cualquier mensaje que llegue hasta el nodo debe pasar por los módulos de radio (transmisor/recepción) y MAC antes de determinar si el mensaje corresponde a este nodo o si se debe reenrutar el mismo haciéndolo pasar nuevamente por los módulos de MAC y radio. Las capas ya desarrolladas de MAC y radio se encargan solo de pasar el mensaje hacia el siguiente submódulo dentro del módulo global de comunicación.

Como se puede observar en la tabla 3.2.1, ya se han implementado enrutamientos de tipo *Bypass* en el que los datos pasan a través del módulo sin ninguna consideración especial o el enrutamiento *MultipathRings* en el cual se traslada la posibilidad de reenrutar datos de manera periódica por medio de un tótem en una especie de anillo de comunicación. Ambas formas de enrutamiento no cumplen con los requisitos dados para el proyecto pues el primero no toma ninguna consideración al pasar los mensajes al siguiente submódulo y el segundo posee grandes intervalos de duración y además no recrea el comportamiento de los nodos CRTECMOTE. Es por estas razones que la forma de enrutamiento de los nodos simulados en Castalia debe ser reestructurada.

Ahora bien, se pensó en un principio implementar para el enrutamiento que emulara los nodos CRTECMOTE aplicando en el módulo routing un forma de enrutado de los datos utilizando la configuración source-sink, o fuente-sumidero, en la cual se emplean los multi-saltos. Sin embargo, si se implementara esta solución y a la vez se optará por un módulo MAC con el estándar IEEE 802.15.4 se tendría el inconveniente que los beacon generados no pueden y no deben ser reenrutados pues esta función no le correspondería a la capa MAC lo que generaría que los nodos fuera del alcance del coordinador no se asociaran con la PAN nunca y por ende quedaran fuera de la red.

Para solucionar este inconveniente se procedió, como se verá en la siguiente sección, en trasladar las funciones de enrutamiento a la capa MAC, funciones permitidas por el estándar, y dejar el módulo routing en modo *Bypass* solamente.<sup>3</sup>

### 5.2.3. Módulo de Media Access Control (MAC)

Entre los módulos de control de acceso al canal que ya se han desarrollado para Castalia se encuentra el módulo de protocolo de comunicación según el estándar IEEE 802.15.4 [7]. Este módulo fue implementado con las siguientes especificaciones [5]:

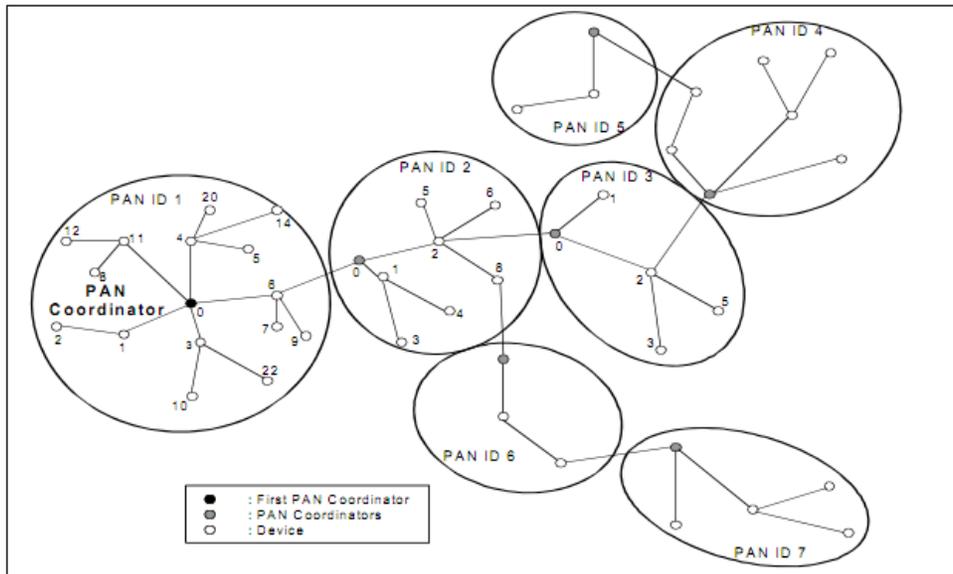
- Funcionalidad CSMA-CA (segmentado y sin segmentar)
- Asociación a PAN por medio de beacons (auto asociación)
- Transmisión de datos de forma directa
- Guaranteed time slots (GTS).

Sin embargo, no se han implementado la generación de beacons secundarios y el flujo de información a través de diferentes PAN (multi-salto) por lo que el módulo debe reestructurarse para adaptarlo al proyecto. Como se mencionó en la sección anterior, la generación de PAN secundarias es indispensable pues se trasladarán las funciones de enrutamiento a la capa MAC.

Para la generación de redes multi-salto en la MAC se tomó como base la caracterización de los dispositivos como función entera (FFD) y función reducida (RFD) dentro de la red según el protocolo IEEE802.15.4. En base a esta clasificación se asignó a todos los dispositivos con función entera la generación de un beacon secundario sincronizado al beacon principal el cual se genera por un dispositivo con función entera que haya sido denotado como coordinador principal de PAN, creando así una red en topología de árbol de grupos o *cluster tree* justo como se observa en la figura 5.2.4

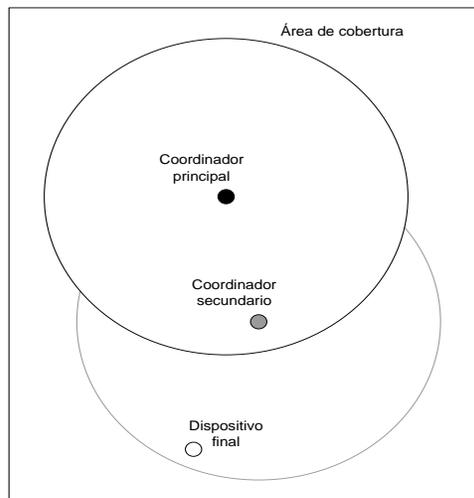
---

<sup>3</sup> Decisión tomada en conjunto con el asesor del proyecto.



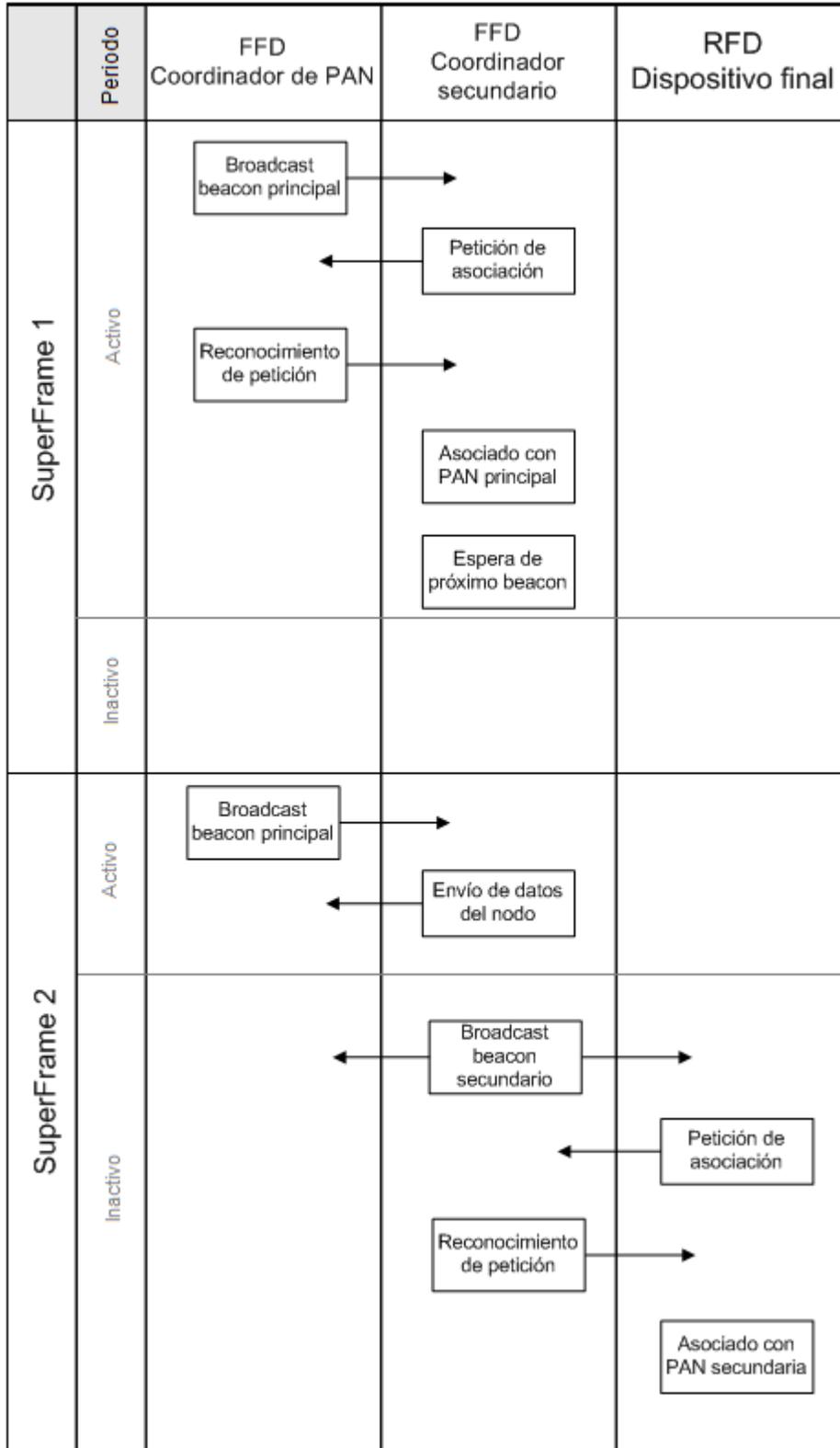
**Figura 5.2.4** Topología de red de árbol de grupos aplicada al proyecto [7]

En la figura 5.2.5 se presenta una configuración básica de árbol de grupos con solo los tres elementos básicos de la misma: un dispositivo FFD trabajando como coordinador principal de PAN, un FFD funcionando como coordinador de PAN secundaria y un dispositivo final (RFD).



**Figura 5.2.5** Configuración básica de árbol de grupos

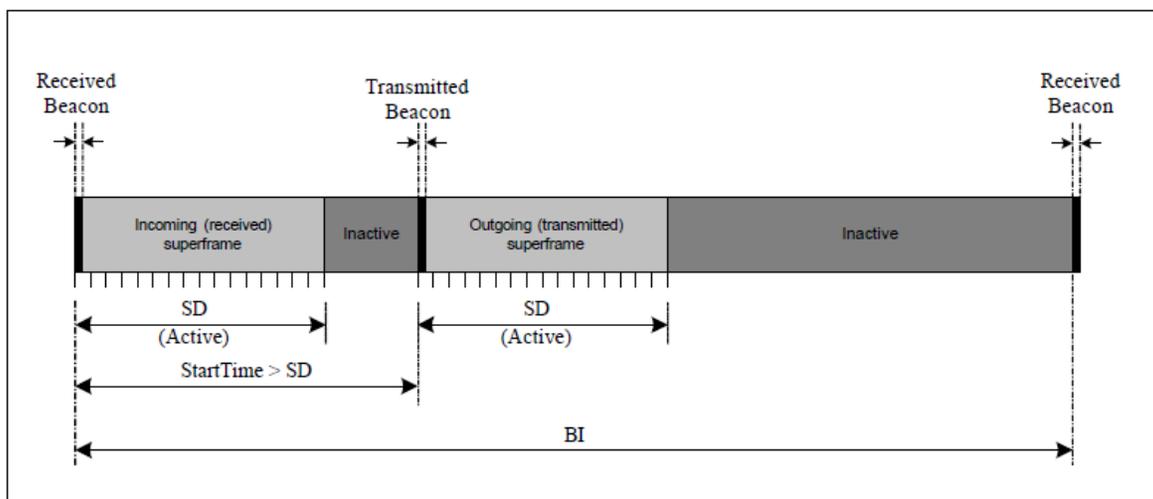
La creación de la red para la figura 5.2.5 se realiza mediante el proceso que se presenta en la figura 5.2.6.



**Figura 5.2.6** Proceso de asociación con tramas guía a PAN principal y PAN secundaria

Cabe destacar que esto se cumple para todos los dispositivos que presenten alguna de las tres funciones descritas en la figura 5.2.6. Además, se debe enfatizar que el proceso descrito en esta figura parte del supuesto que el dispositivo final (RFD) se encuentra fuera del área de alcance del nodo que funciona como coordinador principal, como se observa en la figura 5.2.5, por lo que se necesita de un coordinador secundario para asociarlo a la red. En el caso que el nodo RFD se encontrará dentro del área de cobertura del nodo central (también llamado sumidero), este se asociará a la PAN principal de igual manera que lo hizo el coordinador secundario.

Entrando en detalle en el proceso propuesto por la figura 5.2.6, se puede observar que la generación de beacon secundarios, y por ende de PAN secundarias, se da en el periodo inactivo del super encapsulado principal. Esta solución fue implementada siguiendo el estándar IEEE 802.15.4 [7] en relación con la generación de tramas guía de entrada y salida. De esta forma, dentro del periodo inactivo del super encapsulado principal se dará la creación de un super encapsulado secundario, sincronizado por medio de los tramas guía principales de manera que no exista interferencia entre ambos. Este concepto se presenta gráficamente en la figura 5.2.7.



**Figura 5.2.7** Relación entre beacon de entrada (principal) y beacon de salida (secundario) en un nodo FFD [7]



Se debe recordar que los nodos en Castalia son únicos, o sea que no hay distinción o módulos diferentes entre nodos FFD y RFD, por lo que esta diferenciación debe hacerse por medio de los parámetros de inicialización de cada nodo en Castalia.

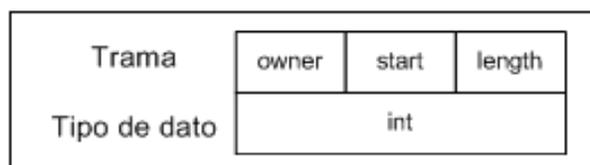
En la figura 5.2.8 se muestra como condición en algunos puntos el mantenerse en el tiempo activo. En el caso de los super encapsulados secundarios se debe comprender que el tiempo del periodo activo no es el mismo que el tiempo del periodo activo del super encapsulado principal pues de ser así ambos encapsulados interferirían entre sí. Para evitar esto, el encapsulado activo secundario debe ser más pequeño. La forma de conseguir esto es por medio del fraccionamiento del parámetro *frame Order* el cual, como se observa en la ecuación 3.3.1, determina el intervalo del tiempo activo. Generalmente este fraccionamiento se realiza entre número que sean múltiplos de dos. En el caso de este proyecto se escogió un fraccionamiento base de entre 4 para evitar problemas de interferencia pues los encapsulados principales en aplicaciones de sensado son muy cortos.

Ahora bien, el nodo FFD funcionando como coordinador principal calcula el intervalo de beacon y del super encapsulado por medio del *frameOrder* y el *beaconOrder* pero, ¿cómo conocen estos parámetros los demás nodos? La respuesta dada por el estándar es agregar en el encabezado de trama de todos los beacon los valores del *beaconOrder* y el *frameOrder* de modo que todos los nodos puedan realizar los cálculos de intervalos de tiempo. Este encabezado es implementado por Castalia utilizando los archivos tipo mensaje (extensión .msg) de Omnet++ con el formato presentado en la figura 5.2.9.

Trama	Mac802154PacketType	PANid	srcID	dstID	nextHop	numberHops	beaconOrder	frameOrder	CAPlength	GTSlength	GTSlist
Tipo de dato	int										GTSspec

**Figura 5.2.9** Formato de encabezado de tramas tipo beacon en Castalia

El espacio GTSlist del encabezado de trama representa los espacios o *slots* garantizados en el CFP y representados en Castalia por una estructura con el formato de la figura 5.2.10, llamada GTSspec.



**Figura 5.2.10.** Estructura GTSspec

En el encabezado presentada en la figura 5.2.9, se presentan dos espacios relacionados a la función multi-salto que se traslado a este módulo desde la capa de enrutamiento: *nextHop* y *numberHops*. Estos dos parámetros permiten determinar cual es el siguiente paso en el flujo de información multihop que para efectos de este proyecto será la dirección de PAN a la que el nodo esta asociada, pues las redes CRTECMOTE siempre tendrán una distribución fuente-sumidero, y el número de saltos que el mensaje a realizado, respectivamente. En relación con el número de saltos, se debe recordar que el estándar IEEE 802.15.4 permite hasta un máximo de 4 saltos, lo cual se regulará con el parámetro *numberHops*.

Seguidamente, se contempla el manejo de potencia en la solución. Como se mencionó anteriormente (sección 5.2.1) el módulo MAC corresponde junto con los módulos de aplicación y de enrutamiento al consumo de potencia que tendrá el microcontrolador. De la misma manera que el módulo de aplicación, este módulo realizará en su inicialización una lectura de un archivo de parámetros de consumo (archivo de tipo texto) con un formato como el presentado en la figura 5.2.3. A partir de estos valores decidirá cual es la potencia que consumirá en estado activo o normal (envío y recepción de mensajes) o en estado de reposo.

Los módulos de aplicación, enrutamiento y MAC tendrán un vínculo fuerte en términos de manejo de potencia. Inicialmente, el nodo iniciará en un consumo de potencia *idle* o de reposo y no pasará al estado de consumo normal hasta que

alguno de los módulos entre en funcionamiento. El nodo volverá al estado de consumo idle solamente cuando los tres módulos se encuentren en reposo. Esto se debió tener en consideración en el módulo de manejo de recursos, como se verá más adelante.

#### 5.2.4. Módulo Radio

Se mencionó previamente que la solución implementada por los desarrolladores de Castalia para la parametrización de los dispositivos de transmisión/recepción consistió en la utilización de archivo de texto lo que le da más versatilidad al simulador. Esta solución se ajusta perfectamente a lo necesitado en este proyecto.

Sin embargo, entre los dispositivos que se han parametrizado en archivos de texto para Castalia no se encuentra el transmisor/receptor MRF24J40 utilizado por los nodos CRTECMOTE. Para adaptar este dispositivo a las simulaciones se procedió a parametrizar el mismo basándose en la hoja de datos aportada por el fabricante [9] y colocarla como una más de las opciones de dispositivos a escoger por el módulo Radio (en carpeta de parámetros de Radio en Castalia). En la tabla 5.2.1 se resumen los parámetros usados en el archivo de parametrización.

**Tabla 5.2.1.** Parámetros usados en archivo de parametrización de Castalia para dispositivo MRF24J40

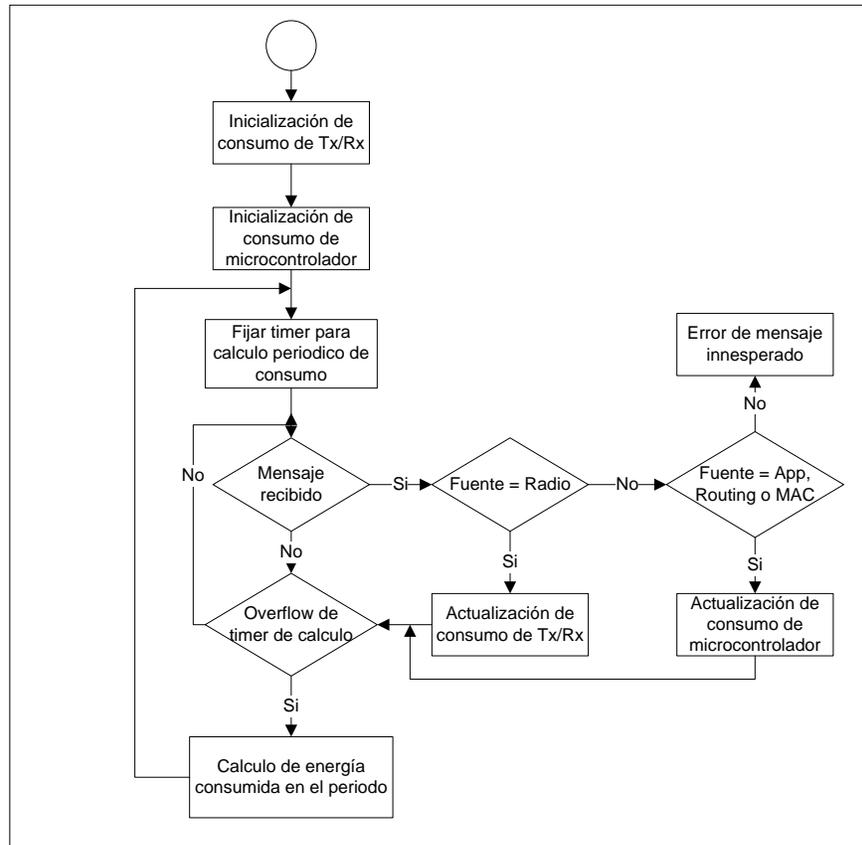
Modo	Parámetro	Valor
Recepción	Potencia consumida	62,7 mW
	Sensitividad	-94 dBm
	Data Rate	250 kbps
	Frecuencia de operación	2400 MHz
	NoiseFloor	-132 dBm
	Bits por símbolo	4
Transmisión	Potencia consumida	75,9 mW
Reposo	Potencia consumida	0,0165 mW

### **5.2.5. Módulo de manejo de recursos**

Este módulo realiza una especie de “contabilidad” de los recursos de potencia y memoria con los que cuenta el nodo para su funcionamiento. No posee equivalente en la realidad pues este representa funciones abstractas, por ejemplo si el nodo se quedará sin energía no existe un módulo que contabilice este hecho en la realidad, sencillamente dejaría de trabajar por falta de recursos. Los datos recopilados por el módulo de manejo de recursos sirven para evaluar el rendimiento de un nodo dentro de la red y es determinante para la evaluación global de la misma.

En la última versión de Castalia los desarrolladores solo habían incorporado el consumo de energía del transmisor/receptor del nodo en el módulo de manejo de recursos. Debido a la necesidad de obtener el consumo real del nodo, incorporando todos sus componentes, es necesario agregar al manejo de recursos de potencia los módulos de aplicación, enrutamiento y MAC que representan, como se pudo observar en la figura 5.1.1, el microcontrolador. Cada uno de estos módulos, al cambiar de modo de operación envía un mensaje directo al módulo de manejo de recursos.

El funcionamiento del módulo de manejo de recursos se basa en mantener una contabilidad del consumo de potencia de los módulos de aplicación, enrutamiento, MAC (estos tres como un conjunto) y radio. Este proceso se resume por medio del diagrama de flujo de la figura 5.2.11.



**Figura 5.2.11.** Diagrama de flujo del funcionamiento del módulo de manejo de recursos

Al inicializar los nodos cada módulo inicia en modo de consumo de reposo hasta que deba efectuar alguna tarea. Al realizar la primera tarea, y por ende al cambiar al modo de consumo normal, se envía un mensaje al módulo de manejo de recursos informándole de este cambio. El módulo de manejo de recursos interpreta cual es la fuente del mensaje y cambia el modo de consumo para este módulo para luego, periódicamente, calcular la energía promedio consumida por intervalos. Se debe enfatizar que los mensajes recibidos de los módulos de aplicación, enrutamiento y MAC son manejados como iguales pues representan el mismo dispositivo en la realidad. Si alguno de éstos pasa de modo idle a modo de consumo normal, este estado no se restablecerá hasta que los tres módulos se encuentren en reposo. Una vez finalizada la simulación se resume el consumo de potencia en términos de los Watts consumidos en promedio y los Joules consumidos de la energía inicial del nodo.

En el caso del manejo de memoria, esto representa el espacio requerido por los datos almacenados en los buffer y de la misma manera que el consumo de potencia se realiza mediante el envío de mensajes una vez que se ocupen o desocupen los espacios de la memoria de los módulos. Esta parte del módulo funciona apropiadamente para los requerimientos del proyecto por lo que no debe ser reestructurada.

### **5.3. Relación entre módulos dentro de nodos en Castalia**

Ya se han descrito las conexiones existentes entre los módulos dentro de los nodos en Castalia (véase figura 3.2.3). Ahora se deben de enfatizar las funciones creadas para manejar el flujo de información a través de estas conexiones. En el modelo de capas de comunicación OSI se utiliza el empaquetado de los datos uniéndolos a un encabezado apropiado (según la capa) para enviarlos hacia la capa inferior. De igual manera, al recibir un mensaje de una capa inferior el mismo se debe desempaquetar (eliminar el encabezado). Omnet++, como ya se mencionó, sigue el mismo concepto para el manejo de mensajes entre módulos. En la tabla 3.1.1 se mencionaron las funciones utilizadas por la herramienta para estas comunicaciones.

Dentro de las clases auxiliares enunciadas en la sección 5.1 se encuentran varias funciones básicas para el manejo de datos entre módulos, de las cuales las más importantes se mencionan a continuación:

- `VirtualApplication::toNetworkLayer(cMessage * msg)`
- `VirtualApplication::toNetworkLayer(cPacket * pkt, const char *dst)`
- `VirtualRouting::toApplicationLayer(cMessage * msg)`
- `VirtualRouting::toMacLayer(cMessage * msg)`
- `VirtualRouting::toMacLayer(cPacket * pkt, int destination)`
- `VirtualMac::toNetworkLayer(cMessage * macMsg)`
- `VirtualMac::toRadioLayer(cMessage * macMsg)`

Las funciones que no se hayan mencionado en la lista anterior deberán ser desarrolladas en los módulos específicos como el Mac 802.15.4 o el TMAC en caso de la capa MAC.

#### **5.4. Cálculo de pérdidas por distancia en comunicaciones**

En el modelo general de las redes de sensores en Castalia, los nodos se interconectan utilizando el módulo *Wireless Channel* para representar el medio de propagación de los datos, el aire generalmente. Dentro de este módulo se calculan la pérdida que experimenta la señal entre nodo y nodo al pasar por el canal basándose en potencia de transmisión y distancias entre los nodos. Castalia se encuentra en la actualidad como uno de los simuladores más realistas en la simulación de las pérdidas producidas por el canal pero para éstas seas fidedignas, se debe escoger el método apropiado para el cálculo de dichas pérdidas. La herramienta posee dos formas básicas para el cálculo de las pérdidas en el aire [5]:

- Modelo de sombra lognormal en una dirección
- Modelo de sombra lognormal bidireccional
- Archivo de mapa de pérdida por trayectoria

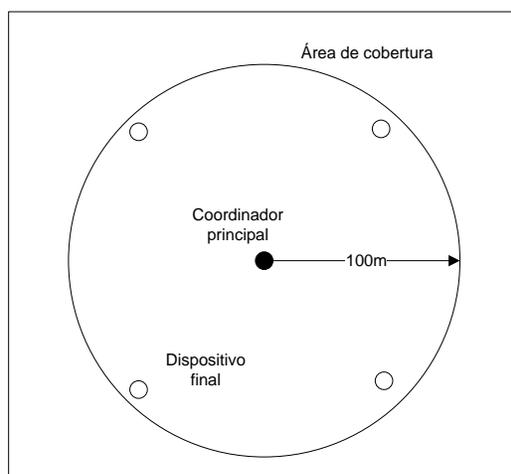
Los dos primeros métodos enunciados funcionan para redes en donde la distancia entre los nodos sea desde uno pocos hasta unos cientos de metros, esto en espacios cerrados o áreas altamente pobladas pero no funcionan de manera adecuada en casos en los que los nodos estén separados por distancias de menos de un metro, como por ejemplo las redes de área corporal. Para las redes inalámbricas de sensores, sin embargo, el modelo de sombra lognormal sirve de manera certera. Este modelo se basa en el cálculo de la ecuación de pérdida por distancia expresada en la ecuación 5.4.1.

$$PL(d) = PL(d_o) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_o}\right) \cdot X_\sigma \quad (5.4.1)$$

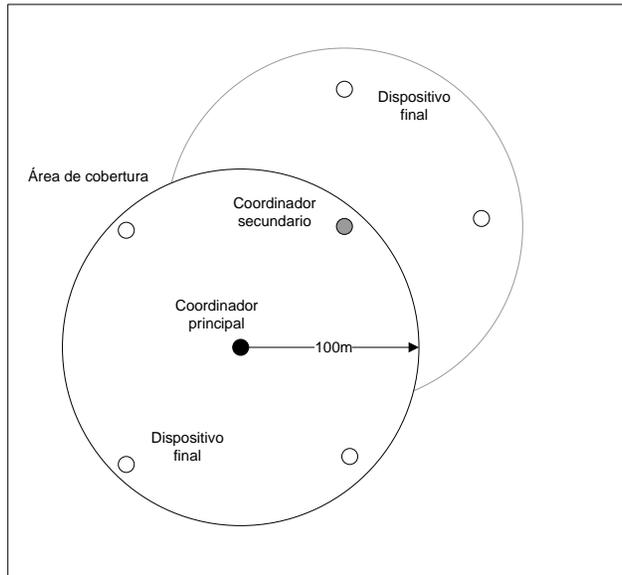
En dicha ecuación,  $PL(d)$  representa la pérdida por trayectoria del nodo,  $PL(d_o)$  una pérdida por trayectoria conocida a una distancia  $d_o$  usada como referencia,  $\eta$  el exponente de pérdida por trayectoria (igual a 2 al aire libre) y  $X_\sigma$  un número al azar dentro de la distribución gaussiana normal con desviación estándar igual a  $\sigma$ . De este modo, el módulo de canal inalámbrico de Castalia puede calcular la pérdida por trayectoria entre los nodos y pasar dicho dato al módulo de radio para que este determine si la señal llega al nodo con intensidad sobre o por debajo de la sensibilidad del mismo.

## 5.5. Implementación de configuraciones de prueba y configuración de red base

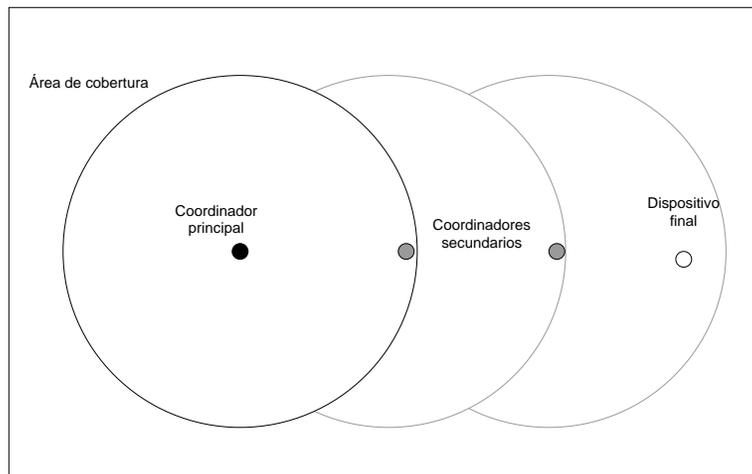
Para corroborar el correcto funcionamiento de los cambios realizados se debieron ejecutar algunas configuraciones de prueba. Estas se muestran en las figuras 5.5.1, 5.5.2 y 5.5.3. Debido a la necesidad en esta etapa de comprobar solo la conectividad de los nodos, no se incorporó información a las tramas de datos.



**Figura 5.5.1.** Configuración de prueba para conectividad



**Figura 5.5.2.** Configuración de prueba para generación de PAN secundaria

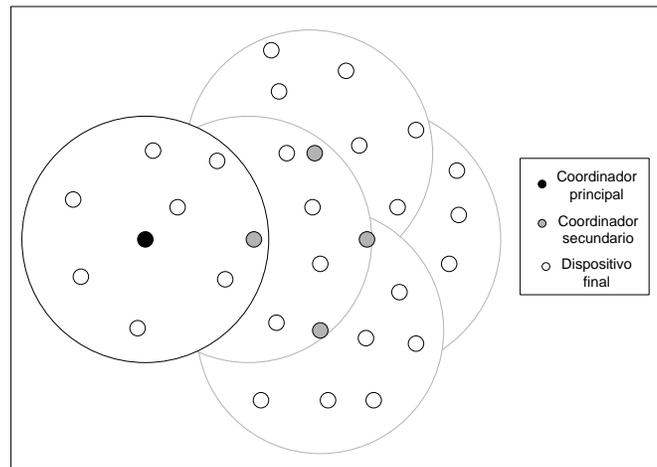


**Figura 5.5.3.** Configuración de prueba para multi-salto

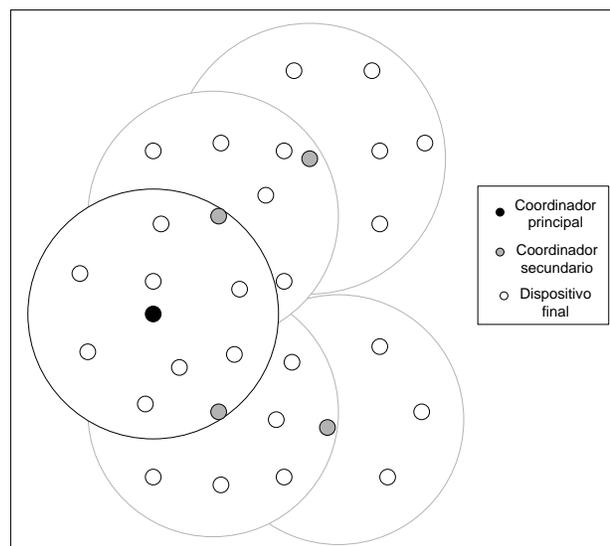
Seguidamente, se procedió a implementar la red base. En la figura 3.4.1 se mostró la configuración implementada en la realidad con nodos CRTECMOTE en proyectos anteriores. Gracias a esta implementación ya se cuenta con valores estadísticos de funcionamiento de la red por lo que se procedió a utilizarla como red base del proyecto.

## 5.6. Implementación de configuraciones complementarias

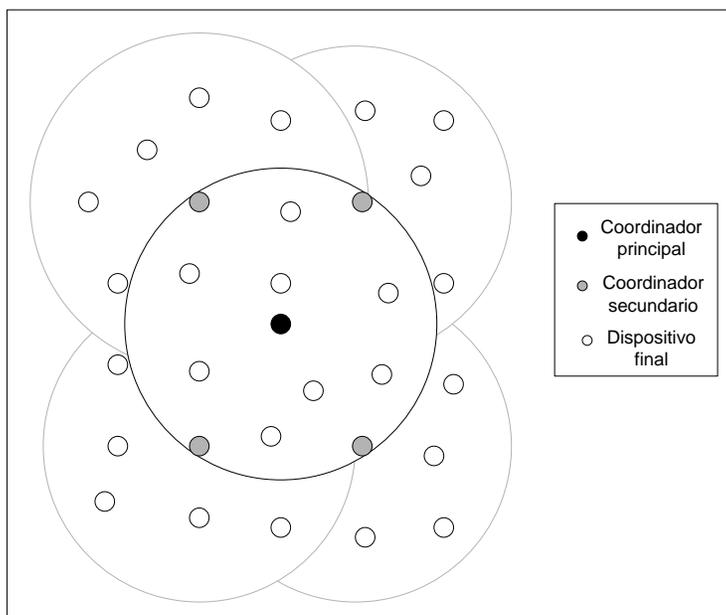
Luego de que se ejecutó la simulación de la configuración base del proyecto se procedió a ejecutar configuraciones complementarias buscando la configuración de red con mayor rendimiento. Las configuraciones que se simularán se presentan en las figuras 5.6.1, 5.6.2, y 5.6.3.



**Figura 5.6.1.** Configuración compleja de red con nodo sumidero en el extremo y una sola PAN secundaria convergente



**Figura 5.6.2.** Configuración compleja de red con nodo sumidero en el extremo dos PAN secundarias convergentes



**Figura 5.6.3.** Configuración compleja de red con nodo sumidero en el extremo y cuatro PAN secundarias convergentes

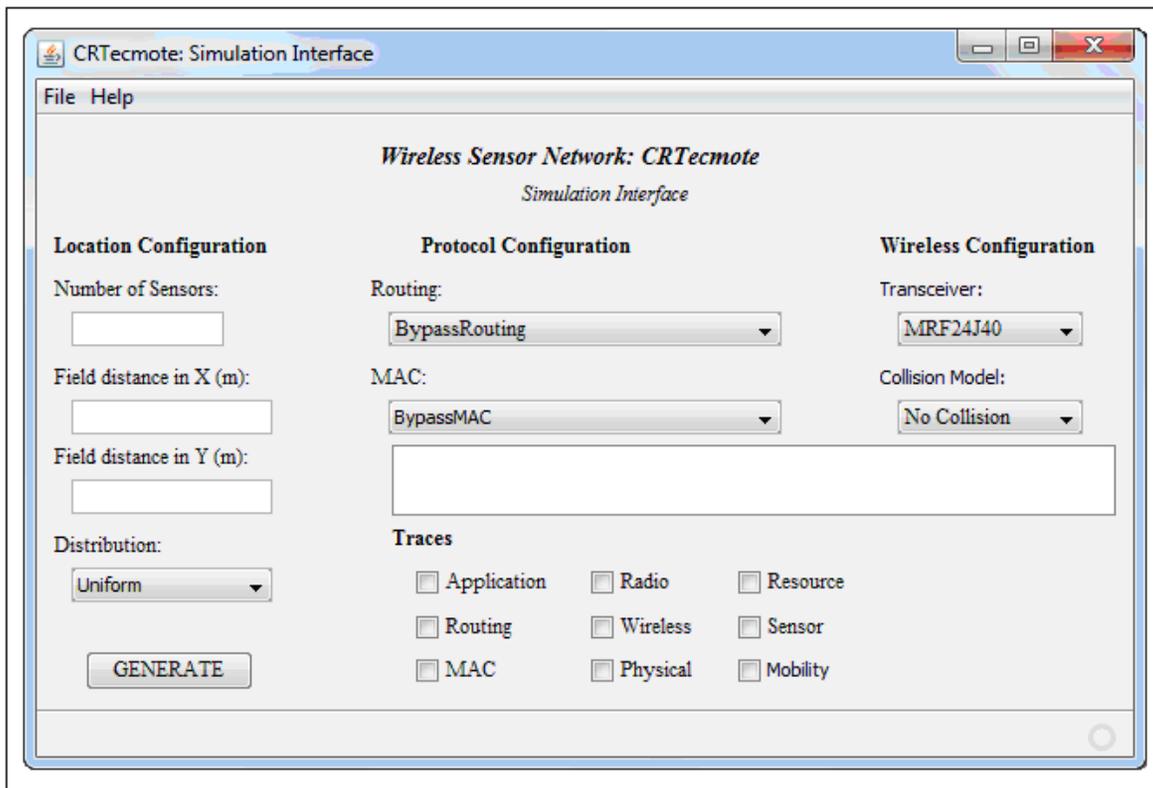
Estas configuraciones fueron seleccionadas para comparar redes en donde el nodo sumidero se encuentra en el extremo de la red con diferentes PAN secundarias convergentes, una red con una única PAN convergente y el nodo sumidero en la misma posición y aquella donde el nodo sumidero se encuentra en el centro de la red. Dichas configuraciones fueron escogidas pues representan la mayor posibilidad de ser aplicadas en la realidad y, en conjunto con el asesor del proyecto, fueron definidas como la mejor opción para las pruebas.

En busca de, no solo de probar la red, sino de forzarla al máximo las tres configuraciones fueron aplicadas con un número alto de nodos (31) en donde solamente se decidió la posición de los nodos FFD y los demás se posicionaron aleatoriamente (comando de Castalia para posicionar nodos al azar pero de manera uniforme), un área relativamente extensa para una red de sensores (200 metros por 200 metro) y un intervalo de lectura de sensores de 1 segundo. Las simulaciones se realizaron en un lapso de un minuto.

## 5.7. Creación de interfaz gráfica

Este proyecto se desarrollo para ser la base de otros avances en el proyecto global con redes CRTECMOTE. Teniendo este pensamiento en mente se buscó facilitar el uso del simulador para los futuros usuarios de Castalia.

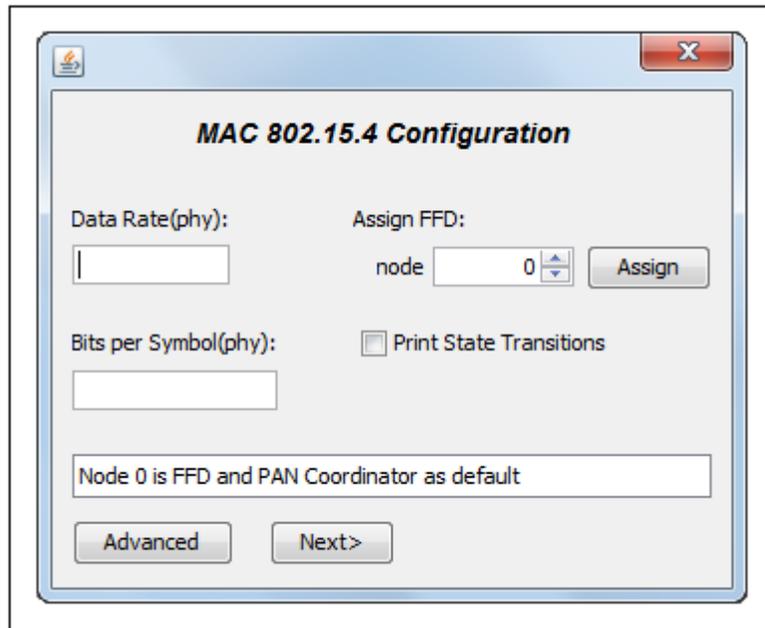
En la figura 5.7.1 se presenta la interfaz de usuario básica implementada por medio de la herramienta JAVA (por medio del compilador Netbeans). Esta interfaz utiliza los valores incorporados a la misma para crear un archivo de inicialización *omnetpp.ini* para Castalia.



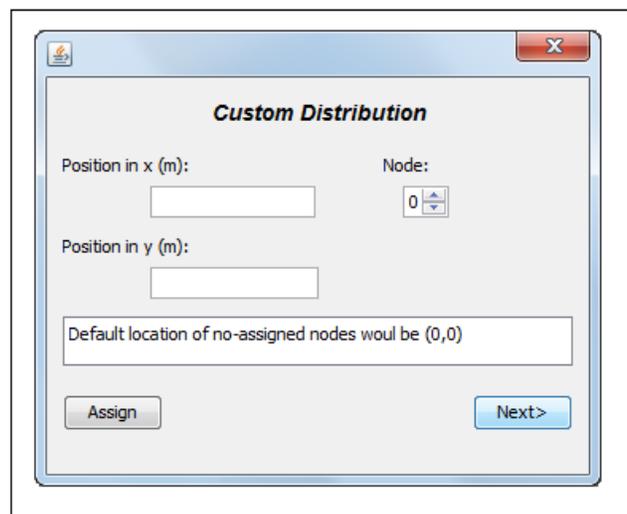
**Figura 5.7.1.** Interfaz gráfica implementada para simulador de nodos CRTECMOTE

Para la implementación del archivo Castalia se toma como defecto la posición del nodo sumidero en el centro de la red y un tiempo de simulación de 60

segundos. Existen dos ventanas emergentes para la elección de los parámetros del protocolo MAC 802.15.4 y la colocación manual de los nodos las cuales se presentan en las figuras 5.7.2 y 5.7.3, respectivamente.



**Figura 5.7.2.** Ventana emergente para configuración del protocolo MAC 802.15.4



**Figura 5.7.3.** Ventana emergente para colocación manual de nodos en la red

## Capítulo 6 Análisis de Resultados

### 6.1. Resultados

#### 6.1.1. Pruebas de funcionamiento

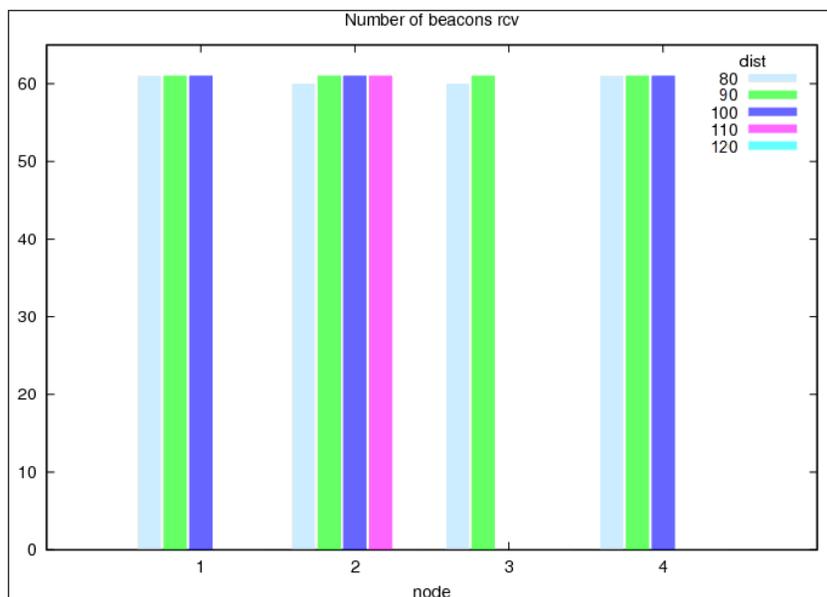
Las pruebas para corroborar el funcionamiento de las reestructuraciones realizadas a Castalia, como se puede observar en las figuras de la sección 5.5 consistieron en tres aspectos básicos: conectividad, generación de redes de área personal secundarias y conexiones multi-salto.

Para corroborar la conectividad de los nodos se ejecutó el escenario de simulación de la figura 5.5.1, en donde un nodo coordinador FFD genera un beacon que debe ser recibido por 4 nodos a su alrededor ubicados a 100 metros de distancia. Este valor representa la distancia máxima de alcance comprobada para un transmisor/receptor [3]. Luego de realizar la ejecución se obtuvieron los resultados de conectividad, tanto de recepción de tramas guía como de paquetes, que se muestra en las tabla 6.1.1. En estas pruebas cada nodo envió 48 paquetes y el coordinador principal FFD generó 62 beacons.

**Tabla 6.1.1.** Tramas guía recibidos por dispositivos finales paquetes recibidos por sumidero para configuración de la figura 5.5.1

Parámetro	Nodo			
	1	2	3	4
Tramas guía	61	60	60	61
Paquetes	47	42	45	46

Para determinar exactamente a qué distancia se pierde conectividad para esta configuración se ejecutó la misma pero con diferentes distancias entre el nodo sumidero y los nodos en el mismo archivo de inicialización para facilitar su comparación. El resultado de esta comparación se presenta en la figura 6.1.1.



**Figura 6.1.1.** Gráfico de tramas guía recibidos por dispositivos finales para configuración de la figura 5.5.1 con diferentes distancias entre nodos

Seguidamente, se procedió a comprobar la generación de redes PAN secundarias sincronizadas a la red principal. Para esto se ejecutó la simulación de la red mostrada en la figura 5.5.2, en donde el nodo de la esquina superior derecha (nodo 4 en términos de la simulación) fue asignado como FFD no coordinador principal. Los resultados de esta simulación se presentan en la tabla 6.1.2. Se debe destacar que para esta prueba, se contemplan aspectos de conectividad prioritariamente. Es importante conocer que el número de beacons principales generados en esta simulación fue de 62, se generaron 60 beacon secundarios y cada nodo envió hacia el sumidero 59 mensajes o paquetes.

**Tabla 6.1.2.** Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.5.2

Parámetro	Nodo				
	1	2	3	4	5
Tramas guía	61	61	60	61	60
Paquetes	58	41	41	57	57

Finalmente, en términos de comprobación de funcionamiento, se simuló la configuración de la figura 5.5.3 para corroborar que la red tenga la funcionalidad multisalto. Los resultados de conectividad de esta prueba se presentan en la tabla 6.1.3, en donde el nodo con función de coordinador principal generó 62 tramas guía, el coordinador secundario inmediato al principal (nodo 1 en la simulación) creó 60 beacon secundarios, el segundo coordinador secundario (nodo 2) envió 59 beacons y todos los nodos enviaron al sumidero 59 paquetes de datos.

**Tabla 6.1.3.** Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.5.3

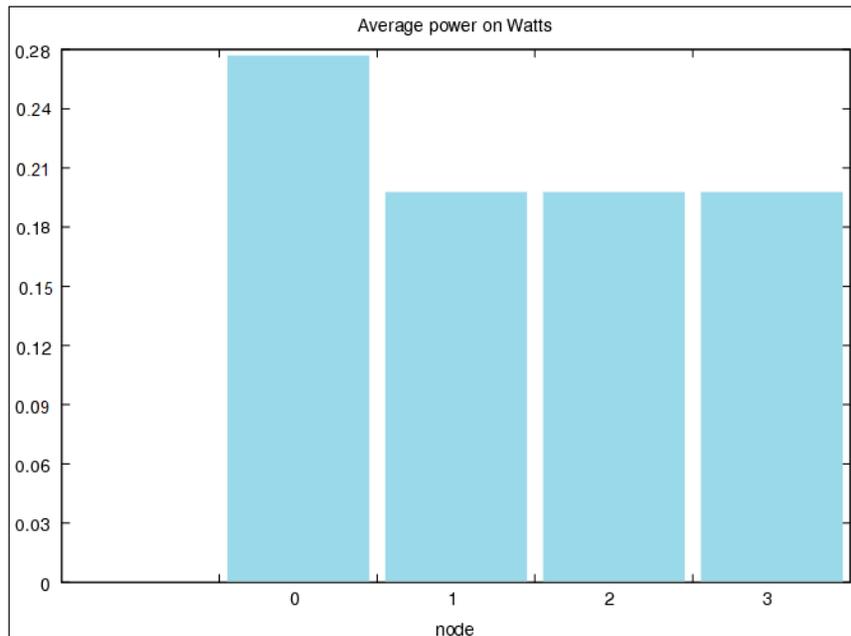
Parámetro	Nodo		
	1	2	3
Tramas guía	61	60	60
Paquetes	59	57	55

### 6.1.2. Funcionamiento de configuración de red base

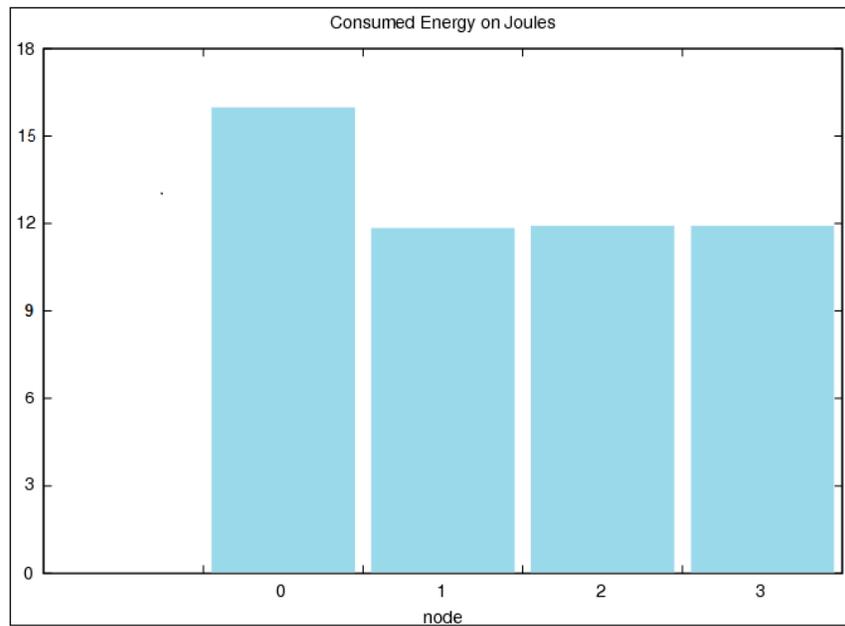
Una vez comprobado el funcionamiento de la reestructuración realizada a Castalia por medio de las pruebas de la sección anterior se procedió a ejecutar la configuración de red utilizada en proyectos anteriores y cuyo esquema se presentó en la figura 3.4.1. Para esta simulación se buscan estadísticas no solo de conectividad, sino también de rendimiento. En la tabla 6.1.4 se presentan los resultados de conectividad mientras que en las figuras 6.1.2 y 6.1.3 se muestran los resultados de consumo de energía. Se debe mencionar además que se generaron 62 beacons por el FFD principal y cada paquete envió 30 paquetes en un tiempo de simulación de 1 minuto.

**Tabla 6.1.4.** Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de red base

Parámetro	Nodo		
	1	2	3
Tramas Guía	61	60	60
Paquetes	30	30	29



**Figura 6.1.2.** Gráfico de energía promedio de nodos para configuración de red base en Watts



**Figura 6.1.3.** Gráfico de energía consumida por los nodos para configuración de red base en Joules

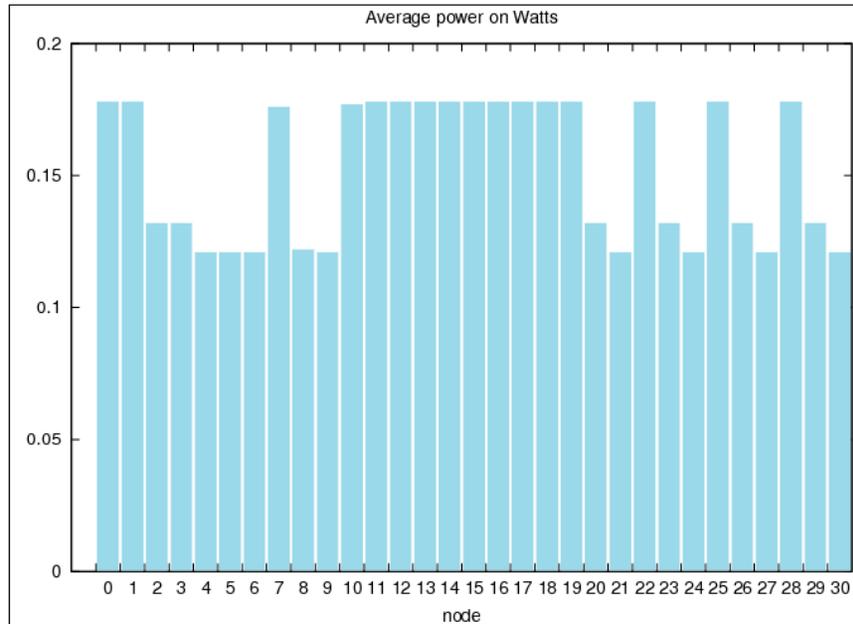
### 6.1.3. Funcionamiento de redes complementarias

En la figura 5.6.1 se presenta la configuración en donde toda la información converge a un nodo por medio de una sola PAN secundaria. Los resultados de número de tramas guía recibidas y paquetes recibidos por el sumidero se resumen en la tabla 6.1.5 mientras que el tiempo de los nodos sin asociación a una PAN, energía promedio de los nodos y energía consumida de estos se presentan las figuras 6.1.4, 6.1.5 y 6.1.6.

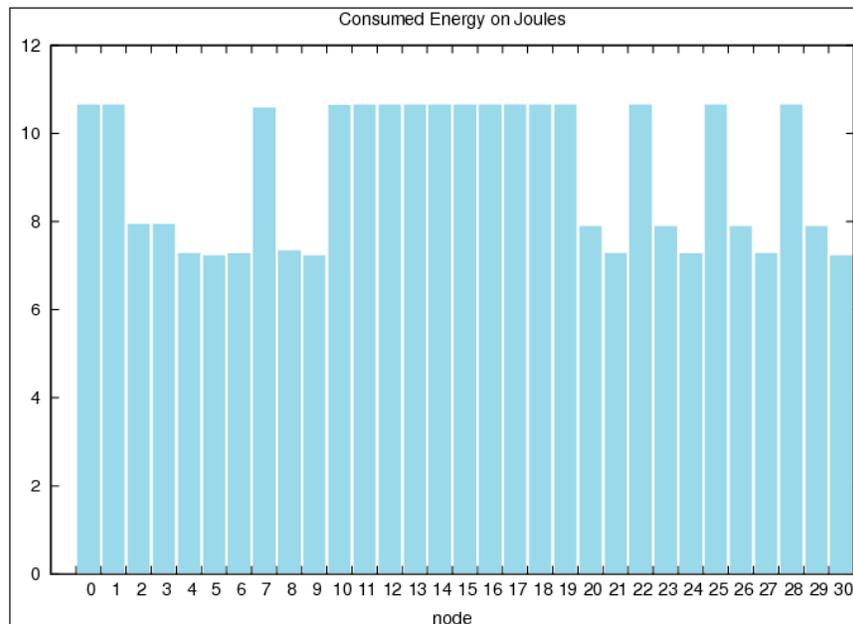
El número de beacons enviados por el FFD coordinador en la configuración fueron 62 en un tiempo de simulación de 1 minuto. Los nodos 1 y 10 son fueron determinados como FFD para generar tramas guía secundarios y generan 60 y 59 faros cada uno. En cuanto a la asociación, el nodo 1 se asoció a la PAN principal y el nodo 10 a la 1. Todos los nodos enviaron 30 paquetes.

**Tabla 6.1.5. Tramas guía** recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.1

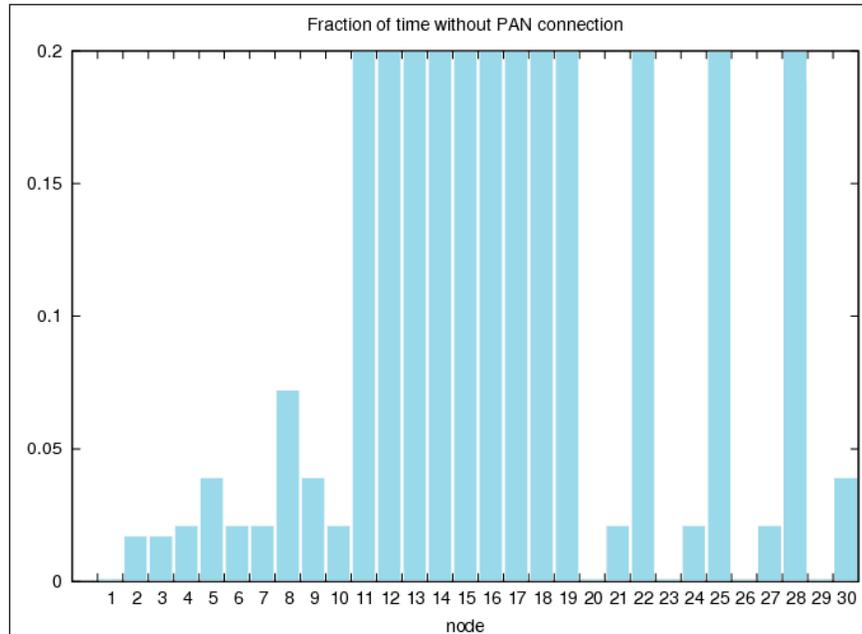
Nodo	Tramas guía	Paquetes	Nodo	Tramas guía	Paquetes
1	61	26	16	0	0
2	60	25	17	60	30
3	60	29	18	59	30
4	60	21	19	0	0
5	60	15	20	61	27
6	60	17	21	60	22
7	60	25	22	0	0
8	62	18	23	0	29
9	60	14	24	60	23
10	60	23	25	0	0
11	59	30	26	61	28
12	0	0	27	60	22
13	0	0	28	0	0
14	0	0	29	61	20
15	0	0	30	60	22



**Figura 6.1.4.** Gráfico de energía promedio de nodos para configuración de figura 5.6.1 en Watts



**Figura 6.1.5.** Gráfico de energía consumida por los nodos para configuración de figura 5.6.1 en Joules



**Figura 6.1.6.** Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.1 en segundos

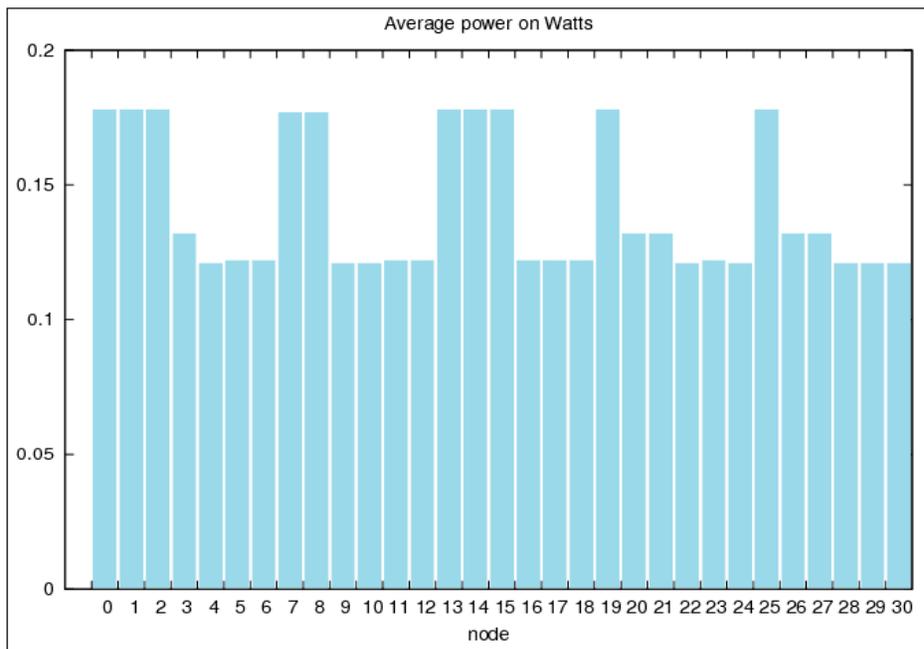
Una configuración similar se presenta en la figura 5.6.2, en donde el nodo se coloca en un extremo de la red pero con la diferencia que la información converge a este por medio de dos PAN secundarias, repartiendo el tráfico de información entre dos nodos. Los resultados para esta simulación se presentan en la tabla 6.1.6 para datos de conectividad y en las figuras 6.1.7, 6.1.8 y 6.1.9 en el caso del consumo de potencia y tiempos sin asociación. El nodo principal genera 62 tramas guía.

En este caso los nodos secundarios 1, 2, 7 y 8 se definieron como FFD de manera que pudieran generar PAN secundarias. Al realizar la simulación el nodo 1 generó 60 beacon secundarios, los nodos 2 y 7 generaron 59 y el nodo restante 58 beacons. Los nodos 1 y 2 se asociaron con la PAN principal mientras los nodos 7 y 8 se asociaron con las PAN generadas por los nodos 1 y 2, respectivamente. Además se puede observar en la tabla 6.1.6 que algunos nodos como el nodo 12 por ejemplo tienen un número menor de beacons recibidos a sus pares en la PAN

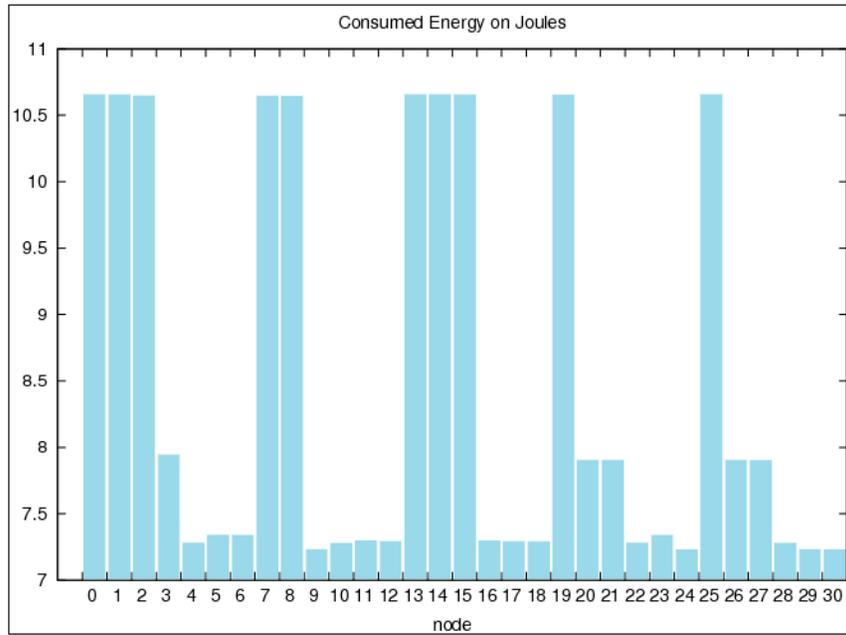
secundaria. En el caso de este nodo se dio un instante en que el mismo perdió contacto con su PAN pero la restableció en el siguiente beacon recibido.

**Tabla 6.1.6.** Tramas guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.2

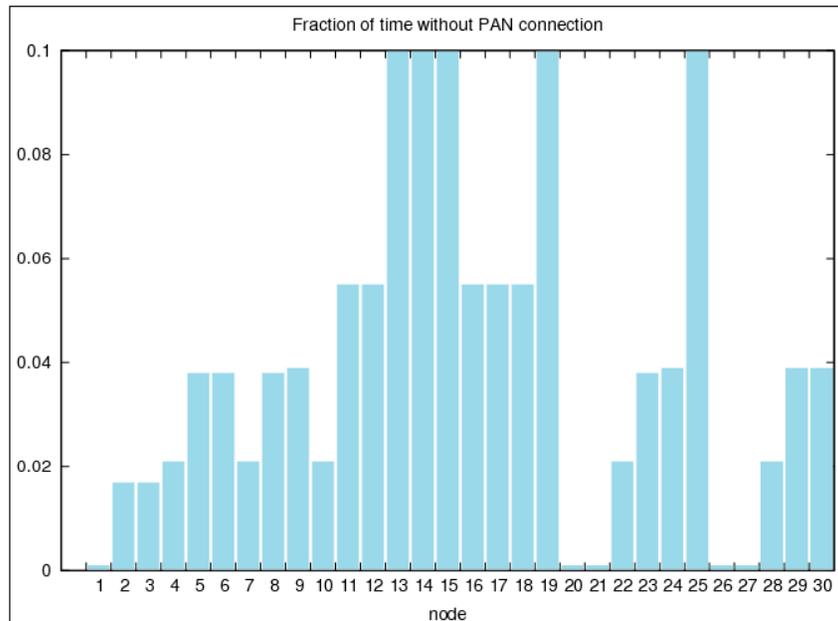
Nodo	Tramas guía	Paquetes	Nodo	Tramas guía	Paquetes
1	61	28	16	58	14
2	60	27	17	58	18
3	60	26	18	58	17
4	60	19	19	0	0
5	59	23	20	61	19
6	59	18	21	61	25
7	60	26	22	60	15
8	59	27	23	59	19
9	59	22	24	59	20
10	60	22	25	0	0
11	58	17	26	61	21
12	58	12	27	61	22
13	0	0	28	60	26
14	0	0	29	61	22
15	0	0	30	59	16



**Figura 6.1.7.** Gráfico de energía promedio de nodos para configuración de figura 5.6.2 en Watts



**Figura 6.1.8.** Gráfico de energía consumida por los nodos para configuración de figura 5.6.2 en Joules

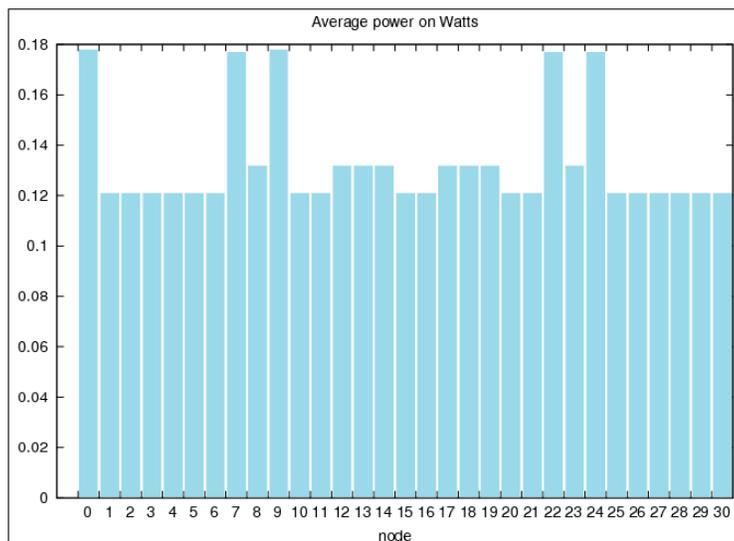


**Figura 6.1.9.** Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.2 en segundos

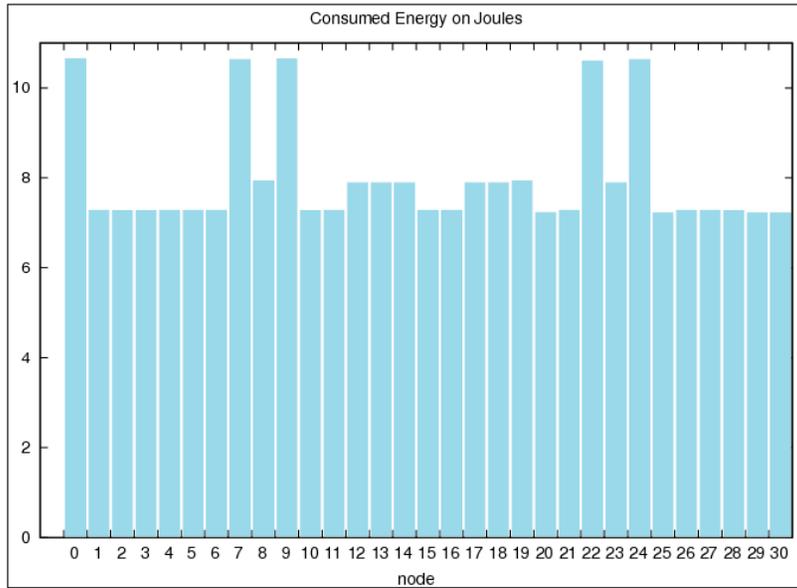
Finalmente, se implementó la configuración presentada en la figura 5.6.3 en donde el nodo sumidero se encuentra en el centro de la red y múltiples PAN secundarias convergen a este. Los resultados significativos, al igual que en los casos anteriores, serán los de conectividad, consumo de potencia y tiempo de falta de asociación, los cuales se resumen en la tabla 6.1.7 y en las figuras 6.1.10, 6.1.11 y 6.1.12.

**Tabla 6.1.7.** Faros guía recibidos por dispositivos finales y paquetes recibidos por sumidero para configuración de la figura 5.6.3

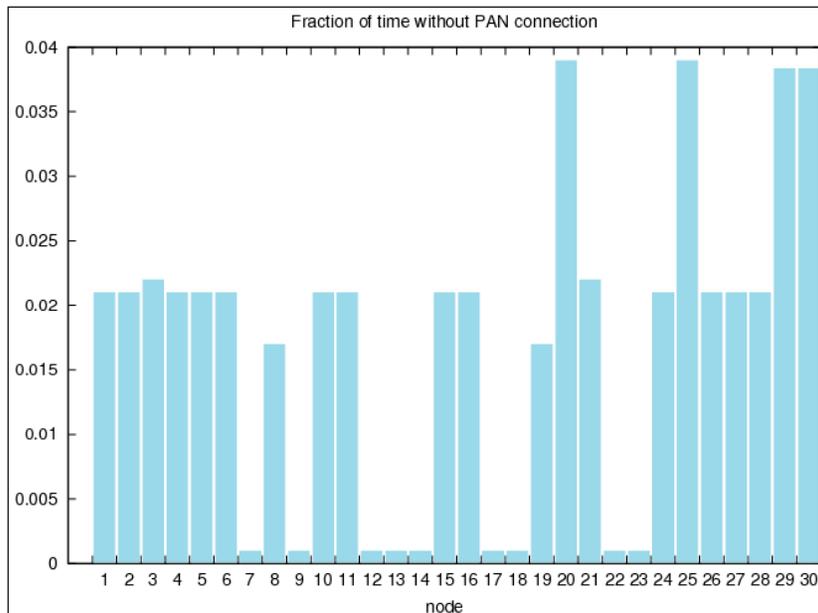
Nodo	Tramas guía	Paquetes	Nodo	Tramas guía	Paquetes
1	60	16	16	60	18
2	60	17	17	61	25
3	61	18	18	61	22
4	60	17	19	60	25
5	60	18	20	59	18
6	60	17	21	60	15
7	61	18	22	61	21
8	60	14	23	61	19
9	61	16	24	60	18
10	60	14	25	59	12
11	61	22	26	60	18
12	61	25	27	60	22
13	61	25	28	61	17
14	61	22	29	59	9
15	60	19	30	59	14



**Figura 6.1.10.** Gráfico de energía promedio de nodos para configuración de figura 5.6.3 en Watts



**Figura 6.1.11.** Gráfico de energía consumida por los nodos para configuración de figura 5.6.3 en Joules



**Figura 6.1.12.** Gráfico de tiempo transcurrido sin conexión a una PAN para nodos de configuración de figura 5.6.3 en segundos

## 6.2. Análisis

### 6.2.1. Conectividad

Este proyecto se orientó hacia la conectividad que las redes inalámbricas de sensores con nodos CRTECMOTE deben lograr para su correcto funcionamiento. Las configuraciones de prueba en la figuras 5.5.1, 5.5.2 y 5.5.3 presentan la base de la conectividad lograda con el simulador para redes de sensores CRTECMOTE.

Con la configuración base de distancia (figura 5.5.1) se buscó comprobar que el sistema tuviera el mismo comportamiento que su análogo real en términos de alcance. Comparando los resultados de esta simulación, mostrados en la tabla 6.1.1 para una distancia de 100 metros y figura 6.1.1 para distancias variables, con los resultados reales encontrados en el proyecto anterior de implementación de nodos CRTECMOTE (tabla 3.4.2) se puede observar que para una distancia de 100 metros se tiene una conectividad similar a la real.

La potencia de recepción de señal para los nodos esclavos en la realidad es menor a -90dBm, mientras que en el simulador esta señal se encuentra en el rango de entre los -90dBm y -93dBm<sup>4</sup>. Estos valores son obtenidos por el simulador restando al valor de potencia de transmisión la pérdida obtenida por medio de la ecuación 5.4.1 (sombra lognormal). Mediante el análisis de esta ecuación se determinó que los valores de pérdida por distancia calculados por el simulador varían según el parámetro  $X_{\sigma}$  que representa una distribución normal alrededor de  $\sigma$ , valor que a su vez varía según el ambiente de simulación (2 para aire libre). Esta variación se ajusta a las variables reales y completamente aleatorias que pueden afectar la conectividad de la red. Se debe enfatizar,

---

<sup>4</sup> Datos tomados de archivos CastaliaTrace.txt generados por Castalia cuya extensión imposibilita adjuntarlos al proyecto.

además, que la base para el cálculo de la señal recibida se tiene la potencia de transmisión. Todo lo anterior apunta a la relación directa que existe entre la conectividad y la distancia entre nodos, la potencia de salida de la señal y variables aleatorias en el medio.

Ahora bien, para comprobar si la distancia a la que los nodos CRTECMOTE pierden la conectividad es simulada apropiadamente por el sistema desarrollado se puede comparar los resultados obtenidos en la figura 6.1.1 con los datos reales en la tabla 3.4.2. En la realidad, los nodos CRTECMOTE empiezan a tener problemas para conectarse con el sumidero cuando la distancia sobrepasa los 100 metros (2 intentos necesarios para la conexión), valor similar a los simulados y presentados en la figura mencionada en donde a 110 metros se inician los problemas de conectividad y para 120 metros, ninguno de los nodos puede asociarse con el coordinador.

En todas las simulaciones se contemplan aspectos de simulación con el resto de parámetros y se puede observar, revisando los resultados generales, que el simulador se comporta apropiadamente para los escenarios desarrollados. Con todo lo anterior, se concluye que el simulador desarrollado posee un comportamiento apropiado en términos de conectividad.

### **6.2.2. Conexiones multi-salto**

El siguiente proceso a analizar en los nodos simulados es la generación de redes de área personal secundarias y la utilización de estas para implementaciones multi-salto. Se debe recordar que las redes de sensores son aplicadas, en su mayoría, en áreas extensas buscando obtener una muestra amplia de datos. Por esta razón es determinante que los nodos CRTECMOTE tengan una manera de enrutar la información de manera adecuada, aún cuando esto represente realizar varios saltos en busca del nodo destino.

Generalmente, los procesos de enrutamiento son realizados por la capa del mismo nombre y emulada por el módulo routing en Castalia. Sin embargo, para unir el enrutamiento en esta capa con el estándar IEEE 802.15.4 para la capa MAC se presentó la limitación antes mencionada para redes con tramas guía, en donde si un nodo no se encuentra asociado a alguna PAN no tiene permisos para transmitir mensaje alguno. Se debe recalcar que los tramas guía no deben ser retransmitidos pues se violaría el estándar.

Analizando detalladamente las opciones se determinó, basándose en parámetros y conceptos establecidos por el estándar, que la mejor manera de implementar los saltos entre nodos de la información es habilitar estas funciones en el módulo MAC. En el proceso de implementación de esta etapa se encontró además que el enrutamiento a nivel de MAC ahorra energía pues la información no debe ser desempaquetada y pasada a la capa en enrutamiento.

En los resultados presentados para las simulaciones de redes con múltiples PAN se encontró que los nodos FFD no pueden ser agregados a la red de manera antojadiza si se busca la optimización en la conectividad de la misma y deben ser colocados de manera que se encuentren en los límites de su propia red principal , esto para obtener el mayor área de cobertura y tratando de no comprometer la conexión de este con una distancia crítica de su nodo coordinador.

Se debe destacar que para la implementación de los multi-saltos se debieron utilizar los espacios para este efecto de la trama de datos. Esto se debió a que otras formas de implementar la determinación del destino de una trama consumen una gran cantidad de energía y no posee la flexibilidad que esta solución aportó.

### **6.2.3. Potencia**

El segundo aspecto más significativo que buscan las redes con nodos CRTECMOTE consiste en los bajos niveles de potencia consumida que deben consumir. Es por esto que las simulaciones de energía deben ser lo más exactas posibles para evitar errores en las implementaciones basadas en simulaciones en Castalia.

En la figura 6.1.2 se presenta la potencia promedio consumida por los nodos en una simulación en Castalia. La misma ejecución fue realizada en la realidad en un proyecto anterior [3] con los resultados mostrados en la tabla 3.4.1. Para los nodos coordinadores se tiene en la realidad una potencia promedio de 0.27273141 mientras las simulaciones aportaron un valor de 0.277634 lo cual presenta un error de solo 1.77% con lo que se demuestra que la solución de utilizar los valores de consumo de cada dispositivo para encontrar el valor promedio de consumo del nodo funciona de manera acertada. En el caso de los nodos esclavos, los valores simulados no pueden ser comprobados pues este valor depende no solo del nodo en sí, sino del consumo de los transductores y otras partes de la parte de sensado los cuales no forman parte del nodo estrictamente y son altamente variables.

Es importante resaltar los datos de potencia consumida en Joules por cada nodo, pues de este parámetro depende la fuente de energía que deberá colocarse a cada nodo para mantenerlo funcionando el tiempo necesitado.

### **6.2.4. Comparación entre redes**

Para determinar la mejor configuración para una red de sensores se optó por comparar redes complejas en donde la diferencia determinante fuera la forma en que la información fluye hacia el sumidero. Estas configuraciones fueron presentadas en las figuras 5.6.1, 5.6.2 y 5.6.3.

Al realizar simulaciones de redes complejas se observó que en algunos casos los nodos esclavos no se conectan a ninguna red, esto porque su posición está fuera del alcance del algún nodo FFD. Cuando esto sucede, el nodo esclavo se mantiene evaluando el medio en busca de una PAN a la cual asociarse lo que consume una cantidad considerable de energía, como se observa en las figuras 6.1.4 y 6.1.7, en donde los valores más elevados de consumo los poseen los nodos FFD y aquellos esclavos que no pudieron lograr la conexión.

En términos de conectividad, se puede apreciar que las tres configuraciones poseen un funcionamiento similar, omitiendo la falta de conexión de algún nodo por distancia, descrita anteriormente. Esto se detalla en las tablas 6.1.5, 6.1.6 y 6.1.7, en donde todos los nodos asociados poseen un número alto de beacons recibidos (de más del 96.7% en el peor de los casos) y en los casos en donde hubiera desconexión recuperan la misma por sí solos.

El flujo de información, por otro lado, no presenta la misma respuesta en los tres casos. Se observó que entre más saltos deba realizar la información para llegar a su destino, menor será el número de mensajes recibidos exitosamente, lo cual se reflejó en los valores promedios de paquetes recibidos. Tomando los valores de recepción de paquetes de las tablas anteriormente mencionadas se calcularon los valores promedio de paquetes recibidos por el nodo sumidero. En el caso de la configuración con una sola red convergente se tiene un promedio de 16.53 paquetes, para la red con dos PAN convergentes se tienen 17.36 mensajes recibidos y para la red con sumidero en el centro se obtuvieron 18.36 paquetes en promedio. Además, se debe mencionar que esta última fue la única configuración que logró conectar todos los nodos en la red y no presentó ninguna desincronización.

Como parte de la solución se decidió no enviar mensajes de reconocimiento de datos adquiridos (acknowledge). La base para realizar esta eliminación consiste en dos razones primordiales: una red de sensores posee un número alto

de paquetes circulando en todo momento y no es apropiado recargarla con datos adicionales y, segundo, en redes de sensores los datos no son determinantes y son tomados en gran cantidad en intervalos cortos de tiempo, por lo que es posible perder datos sin mayor consecuencia.

Es interesante resaltar que la configuración con sumidero en el centro de la red posee 4 nodos FFD colocados en los límites de alcance de la PAN principal. Sin embargo, uno de estos nodos no se asoció a la PAN generada por el sumidero sino a uno de los nodos FFD secundarios y su funcionamiento no se deterioro. Con esto se demuestra la versatilidad que poseen las redes de sensores en Castalia en términos de asociación.

En el caso de la potencia consumida, se puede observar que, en promedio, la configuración con sumidero en el centro presenta el menor consumo. Esto sucede, en parte porque ninguno de sus nodos quedó sin asociación y en parte porque la distribución que presenta hace que se requieran un número menor de saltos para enrutar la información. Se puede afirmar así, que entre más sean las PAN secundarias que convergen hacia el sumidero y entre más uniforme sea la distribución de los nodos alrededor de este, la configuración de la red será más eficiente energéticamente.

## Capítulo 7 Conclusiones y recomendaciones

### 7.1. Conclusiones

1. Una opción acertada para agregar versatilidad al simulador de redes CRTECMOTE en Castalia corresponde al uso de archivos de parametrización.
2. El uso de espacios en el encabezado de la trama facilita la implementación de conexiones multi salto.
3. La conectividad de las redes CRTECMOTE está directamente afectada por la distancia entre los nodos, la potencia de salida de las transmisiones y variables aleatorias externas.
4. Un nodo no asociado a una red de área personal (PAN) consume más energía que un nodo asociado a esta.
5. La parametrización por medio de archivos de texto, el uso del modelo de sombra log-normal y la implementación de conexiones multi-salto del estándar IEEE 802.15.4 para el módulo MAC permiten la apropiada emulación de un nodo CRTECMOTE.
6. Se obtuvo un consumo de energía del nodo sumidero con un 1.77% de error con respecto al valor práctico medido para nodos CRTECMOTE.
7. La configuración óptima para una red inalámbrica de sensores es aquella con el mayor número de PAN convergentes y cuyo sumidero sea colocado en el centro de la red.

## 7.2. Recomendaciones

1. Basándose en el estándar, se tiene un número máximo de 4 saltos permitidos para llegar de un nodo esclavo al sumidero.
2. Los intervalos de sensado pueden interferir en la simulación por lo que se deben definir de modo que sincronicen con los superframe de la capa MAC.
3. Es posible implementar un algoritmo más complejo para el enrutamiento multisalto que no utilice espacios de la trama, pero se sacrificaría tiempo y potencia.
4. Cada vez que se desee ejecutar una simulación se debe calcular que los parámetros frameOrder y beaconOrder sean los adecuados para la misma.
5. Los nodos configurados como dispositivos finales no pueden ser simulados en su totalidad en aspectos de potencia, pues estos dependen directamente del tipo de sensor que posean y del funcionamiento de este

## Bibliografía

[1] Carvajal Godínez, Johan. (2009). Presentación sobre desarrollo de la red inalámbrica en el ITCR, “Diseño de un nodo con arquitectura abierta para redes inalámbricas de sensores: CRTECMOTE”. Instituto Tecnológico de Costa Rica.

[2] Leiva Delgado, Norwin Alexander. (2009). Informe de Proyecto de Graduación de Ingeniería Electrónica, “Sistema de administración en tiempo real de los recursos de hardware y software de un nodo de sensado [SIWA-RTOS]”. Instituto Tecnológico de Costa Rica.

[3] López González, Sebastián. “Diseño e implementación del módulo de comunicación inalámbrica de un nodo CRTECMOTE para enlaces de tipo punto-multipunto”. Instituto Tecnológico de Costa Rica.

[4] Varga, Andrés. “OMNeT++, Discrete Event Simulation System. User Manual.” Diciembre del 2004.

[5] Athanaccios Boulis. Castalia User Manual. NICTA. Marzo del 2011. Descargado en [www.castalia.npc.nicta.com.au](http://www.castalia.npc.nicta.com.au)

[6] Barry, Richard. (2009). Versión 1.0.4. Using the FreeRtos Real Time Kernel, A Practical Guide.

[7] IEEE Std 802.15.4-2006<sup>TM</sup>, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), IEEE, 2006.

[8] Carvajal Godínez, Johan. (2009-2010). Informe final. “Desarrollo tecnológico de un sistema de adquisición de datos ambientales para su uso en proyectos de investigación científica: Arquitectura abierta CRTecMote”. Instituto Tecnológico de Costa Rica.

[9] Microchip Technology Inc. “MRF24J40MA Datasheet – IEEE 802.15.4<sup>TM</sup> 2.4 GHz RF Transceiver (DS39776).”

[10] Microchip Technology Inc. “PIC32MX460F512L Datasheet.” Revisión E, (Julio 2008).

# Apéndices

## A.1 Glosario

**Beacon:** también llamado trama guía, utilizado para sincronización de nodos es una red PAN con protocolo IEEE 802.15.4.

**Encabezado:** datos de enrutamiento agregados al inicio de cada trama.

**FFD:** Dispositivo de funcionalidad completa o entera. Dispositivo capaz de crear una red personal y operar como coordinador.

**Multisalto (conexiones):** conexiones en las que la información pasa por varios nodos dentro de una red antes de llegar a su destino (encabezado).

**PAN:** red de área personal (personal area networks).

**RFD:** Dispositivo de funcionalidad reducida. Dispositivo incapaz de crear una red personal y usado como dispositivo final.

**Trama:** conjunto de bits compuesto por la información a transmitir y los datos de enrutamiento necesarios para que esta alcance su destino.

**WSN:** redes inalámbricas de sensores (wireless sensors Networks).

## A.2 Guía de usuario de Castalia

- **Castalia (input)**

**Castalia -c CONFIG:** Permite ejecutar la configuración determinada en el archivo .ini definido como input. Pueden darse combinaciones de configuraciones uniéndolas por una coma SIN espacio

**Castalia -i FILE:** Selecciona el archivo .ini input, o sea, del cual se leerán las configuraciones. Por defecto es *omnetpp.ini*

**Castalia -o FILE:** Selecciona el archivo en donde se escribirán los resultados de las ejecuciones. Por defecto será un archivo creado a partir de la fecha y hora: *YYMMDD\_HHMMSS.txt*

**Castalia -r N:** Define, por medio de N, el número de repeticiones de un escenario en específico.

- **CastaliaResults (output)**

Se presentan los comandos más importantes de CastaliaResults. Sin embargo, cabe destacar que muchos de estos dependen entre ellos y deben utilizarse en el formato adecuado.

**CastaliaResults -i FILE:** Carga el archivo (generalmente .txt) de resultados

**CastaliaResults -s R:** Muestra una sección de los resultados determinada por RE (-s *packets*, -s *RX*).

**CastaliaResults -n:** Muestra los resultados de cada nodo individualmente

**CastaliaResults -f R:** Filtra la información según el string RE

Un comando de CastaliaResults tendrá un mismo formato básico: escogencia de archivo a abrir, visualización y auxiliares. Por ejemplo:

```
$ CastaliaResults -i 100809-004640.txt -s packets -n
```

- **CastaliaPlot (output)**

Debe tenerse presente que esta aplicación utiliza gnuplot, por lo que debe estar instalado.

**CastaliaPlot -g:** Escala de grises

**CastaliaPlot -l LEG:** Define posición de la leyenda.

**CastaliaPlot -s STYLE:** Define estilo. Opciones: linespoints, histogram, stacked.

**Castalia -o FILE:** Selecciona el nombre del archivo .png de salida.

Otros comandos se pueden encontrar por medio de `CastaliaResults -h`

El formato básico para el uso de `CastaliaPlot` sería:

```
$ CastaliaResults -i 100809-004640.txt -s packets -n | CastaliaPlot -o nombre_archivo.png -s stacked
```

Nótese que primero se deben de usar comandos de `CastaliaResults` para abrir y filtrar la información y luego realizar el gráfico.