



**UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
TECNOLOGIA EN COMPUTACIÓN E INFORMÁTICA**

Monografía

**ANÁLISIS COMPARATIVO DE DIFERENTES
SIMULADORES DE SENSORES INALÁMBRICOS DE
CÓDIGO ABIERTO**

**LÍNEA DE INVESTIGACIÓN
REDES Y ARQUITECTURA DE COMPUTADORAS**

AUTORA

BAQUERIZO MENDEZ ANGELA CRISTINA

BALZAR – ECUADOR

2021



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN E INFORMÁTICA

**ANÁLISIS COMPARATIVO DE DIFERENTES
SIMULADORES DE SENSORES INALÁMBRICOS DE
CÓDIGO ABIERTO**

MONOGRAFÍA

Trabajo de titulación presentado como requisito para la obtención del título de

TECNÓLOGA EN COMPUTACIÓN E INFORMÁTICA

AUTORA

BAQUERIZO MENDEZ ANGELA CRISTINA

TUTORA

LCDA. LOOR CAICEDO GINA MGE.

BALZAR – ECUADOR

2021



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN E INFORMÁTICA

CERTIFICACIÓN DE ACEPTACIÓN DEL TUTOR

Yo, LOOR CAICEDO GINA MGE., docente de la Universidad Agraria del Ecuador, en mi calidad de Tutora, certifico que el presente trabajo de titulación: ANÁLISIS COMPARATIVO DE DIFERENTES SIMULADORES DE SENSORES INALÁMBRICOS DE CÓDIGO ABIERTO, realizado por la estudiante BAQUERIZO MENDEZ ANGELA CRISTINA; ha sido orientado y revisado durante su ejecución; y cumple con los requisitos técnicos exigidos por la Universidad Agraria del Ecuador; por lo tanto se aprueba la presentación del mismo.

Atentamente,

Lcda. Gina Loor Caicedo MGE.
TUTORA

Guayaquil, 27 de marzo del 2020



UNIVERSIDAD AGRARIA DEL ECUADOR
FACULTAD DE CIENCIAS AGRARIAS
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN E INFORMÁTICA

APROBACIÓN DEL TRIBUNAL DE SUSTENTACIÓN

Los abajo firmantes, docentes miembros del Tribunal de Sustentación, aprobamos la sustentación del trabajo de titulación: ANÁLISIS COMPARATIVO DE DIFERENTES SIMULADORES DE SENSORES INALÁMBRICOS DE CÓDIGO ABIERTO, realizado por la estudiante BAQUERIZO MENDEZ ANGELA CRISTINA, el mismo que cumple con los requisitos exigidos por la Universidad Agraria del Ecuador.

Atentamente,

ING. MSc.

PRESIDENTE

ING. MSc.

EXAMINADOR PRINCIPAL

LCDA. LOOR CAICEDO GINA MGE.

EXAMINADOR PRINCIPAL

ING. MSc.

EXAMINADOR SUPLENTE

Guayaquil, 27 de marzo del 2020

Dedicatoria

Dedico este trabajo a Dios, a mis padres, a mi esposo, a mis hijos y familia quienes siempre me han motivado y dado las fuerzas para salir adelante y obtener este título que con gran esfuerzo y dedicación lo he podido conseguir.

Agradecimiento

A Dios por estar siempre conmigo en todos los momentos de mi vida por darme inteligencia y sabiduría para resolver los problemas que se me han presentado en mi camino.

Además, agradezco de la manera más sincera:

A las autoridades de la Universidad Agraria del Ecuador.

Al PhD Jacobo Bucaram Ortiz. Rector Fundador de la Universidad.

A la PhD Martha Bucaram de Jorgge, Rectora de la Universidad.

Al PhD Javier Del Cioppo Morstadt, Vice-Rector de la Universidad.

A la MSc. Emma Jácome Murillo, Decana de la Facultad de Ciencias Agrarias

A los maestros del Programa Regional de Enseñanza Balzar de la Universidad Agraria del Ecuador.

A mi tutora quien me ha guiado en mi trabajo monográfico con profesionalismo y dedicación

A todos mis compañeros y amigos.

Autorización de Autoría Intelectual

Yo, BAQUERIZO MENDEZ ANGELA CRISTINA, en calidad de autora del proyecto realizado, sobre “ANÁLISIS COMPARATIVO DE DIFERENTES SIMULADORES DE SENSORES INALÁMBRICOS DE CÓDIGO ABIERTO”, por la presente autorizo a la UNIVERSIDAD AGRARIA DEL ECUADOR, hacer uso de todos los contenidos que me pertenecen o parte de los que contienen esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autora me correspondan, con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8; 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento.

BAQUERIZO MENDEZ ANGELA CRISTINA

C.I. 0921126793

Guayaquil, 27 de marzo del 2020

Índice general

Portada.....	1
CERTIFICACIÓN DE ACEPTACIÓN DEL TUTOR	3
APROBACIÓN DEL TRIBUNAL DE SUSTENTACIÓN	4
Dedicatoria.....	5
Agradecimiento	6
Autorización de Autoría Intelectual	7
Índice general	8
Índice de tablas	11
Índice de figuras.....	12
Resumen	13
Abstract.....	14
1. Introducción.....	15
1.1 Importancia o caracterización del tema	15
1.2 Actualidad del tema	16
1.3 Novedad científica	16
1.4 Justificación del tema.....	16
1.5 Objetivos.....	17
1.5.1 Objetivo general.....	17
1.5.2 Objetivos específicos	17
2. Aspectos metodológicos.....	18
2.1 Materiales	18

2.1.1 Recursos Bibliográficos.....	18
2.1.2. Materiales e insumos.....	18
2.1.3 Recursos humanos.....	18
2.2 Métodos	18
2.2.1 Modalidad y tipo de investigación.....	18
2.2.2 Tipos de métodos	19
2.3 Marco legal	20
3. Análisis y revisión de la literatura	22
3.1 Simuladores de sensores inalámbricos orientados a la red.....	22
3.1.1 Simulador de sensores inalámbricos Castalia.....	22
3.1.2 Simulador de sensores inalámbricos MiXim	26
3.1.3 Simulador de sensores inalámbricos Pawis	28
3.1.4 Simulador de sensores inalámbricos WSNNet.....	30
3.2 Simuladores - emuladores de sensores inalámbricos	33
3.2.1 Simulador - emulador Tossim	33
3.2.2 Simulador - emulador Cooja.....	36
3.2.3 Simulador - emulador Avrora	39
3.3 Comparación de los simuladores de red de sensores inalámbricos de código abierto	42
4. Conclusiones.....	44
5. Recomendaciones.....	46
6. Bibliografía.....	47

7. Glosario..... 54

8. Anexos 56

Índice de tablas

Tabla 1. Comparación de los simuladores Castalia, Mixim, Pawis, WSNNet, Tossim, Avrora y Cooja.....	56
Tabla 2. Comparación de los simuladores en base al modelado escalabilidad y extensibilidad.....	¡Error! Marcador no definido.

Índice de figuras

Figura 1. Instalación de Simulador Castalia	58
Figura 2. Simulador Maxim.....	59
Figura 3. Estructura del marco de simulación PAWiS	59
Figura 4. Arquitectura de bloques de WSNNet	60
Figura 5. Arquitectura TOSSIM: marcos, eventos, modelos, componentes y servicios	60
Figura 6. Ventana principal del simulador Cooja	61
Figura 7. Ventana/pantalla datos de simulación - Castalia.....	61

Resumen

Este trabajo monográfico está enfocado en el análisis comparativo de diferentes simuladores de sensores inalámbricos de código abierto dividiendo la investigación en tres secciones; la primera detalla las características de simuladores de sensores inalámbricos orientados a la red especificando la estructura y funcionamiento para un buen uso, donde se consideraron los simuladores Castalia, Mixim, Pawis y WSNet; la segunda describe las características de simuladores emuladores de nodos de sensores inalámbricos, considerando para este análisis Tossim, Cooja y Avrora; mientras que la tercera, compara los simuladores de sensores inalámbricos de código abierto descritos en las dos secciones anteriores en base a diferentes métricas para su aplicación en redes de sensores inalámbricos. Se encontró que WSNet, Castalia, Avrora y COOJA son herramientas muy eficientes para grandes modelado de red a gran escala, mientras que MiXiM, Pawis y TOSSIM requieren una mayor cantidad de recursos de computación y tiempo para ejecutar las mismas configuraciones de red.

Palabras claves: Avrora, Castalia, Cooja, Omnet, Tossim, WSNet.

Abstract

This monographic study is focused on the comparative analysis of different open source wireless sensor simulators, dividing the research into three sections; the first details the characteristics of network-oriented wireless sensor simulators specifying the structure and operation for good use, where the Castalia, Mixim, Pawis and WSNNet simulators were considered; the second describes the characteristics of wireless sensor node emulator simulators, considering for this analysis Tossim, Cooja and Avrora; while the third compares the open source wireless sensor simulators described in the previous two sections based on different metrics for application in wireless sensor networks. WSNNet, Castalia, Avrora, and COOJA were found to be very efficient tools for large-scale, large-scale network modeling, while MiXiM, Pawis, and TOSSIM require a greater amount of computing resources and time to run the same network configurations.

Keywords: Avrora, Castalia, Cooja, Omnet, Tossim, WSNNet.

1. Introducción

1.1 Importancia o caracterización del tema

Los sistemas de red de sensores inalámbricos (WSN) tienen un gran potencial para traer nuevos entornos de detección inteligente y reemplazar los sistemas de sensores cableados tradicionales con soluciones más flexibles y menos costosas; sin embargo, a pesar de las muchas actividades de investigación que se realizan en varios centros académicos e industriales en este dominio, el área de WSN aún está abierta, y muchos aspectos relacionados con arquitecturas, protocolos y capas de software aún no están bien definidos y estandarizados.

A diferencia de las redes tradicionales como TCP / IP, no existen patrones de diseño comunes, API estándar y soluciones de hardware que se apliquen a todas las aplicaciones posibles; tal es el caso que en ocasiones se parte desde cero y se requiere un enfoque de diseño individual para satisfacer una demanda específica de la aplicación.

Además de los problemas de diseño como son la gestión de los recursos limitados de hardware presentes en el nodo de red, la energía de batería limitada y la infraestructura de comunicación ad-hoc personalizada, existe una amplia variedad de protocolos de enrutamiento y MAC no estándar disponibles, que complica aún más el proceso de diseño, y por ende dificulta la elección de la solución adecuada que tenga una mejor adaptación al sistema específico que se requiere.

Por estas razones es importante poder explorar y validar los diversos aspectos de un espacio de diseño complejo antes de proceder a la implementación real y el despliegue en el mundo real. Los enfoques más comunes para la evaluación de sistemas WSN incluyen modelado, simulación y creación de prototipos.

1.2 Actualidad del tema

En los últimos años se han desarrollado varios entornos de simulación que ofrecen un conjunto diversificado de instalaciones para modelar, examinar y probar aspectos tales como algoritmos de aplicación, protocolos de comunicación, comportamiento de red cooperativa y técnicas de ahorro de energía.

1.3 Novedad científica

Cuando la creación de prototipos no es práctica, se pueden usar modelos para analizar el sistema tanto analíticamente como a través de la simulación; los métodos analíticos se usan comúnmente para evaluar estáticamente clases de técnicas contra algunos criterios de optimización. Cuando el diseñador está interesado en el comportamiento dinámico del sistema, los modelos se usan generalmente junto con la simulación, que actualmente es una de las técnicas más utilizadas para explorar y evaluar el diseño de Aplicaciones WSN.

1.4 Justificación del tema

Cada herramienta de simulación está diseñada para un propósito específico y tiene un nivel diferente de precisión y adecuación dependiendo de la aplicación de destino. Debido a esto, es esencial tener en cuenta las fortalezas y debilidades de los simuladores disponibles, para poder usarlos adecuadamente de acuerdo con el propósito para el que fueron diseñados

Por esta razón este trabajo monográfico pretende realizar un análisis comparativo de diferentes simuladores de sensores inalámbricos de código abierto, mediante los siguientes objetivos.

1.5 Objetivos

1.5.1 Objetivo general

Analizar diferentes simuladores de sensores inalámbricos de código abierto mediante una recopilación bibliográfica para usarlos adecuadamente según el propósito del diseño.

1.5.2 Objetivos específicos

- Detallar las características de simuladores de sensores inalámbricos orientados a la red especificando la estructura y funcionamiento para un buen uso.
- Describir las características de simuladores-emuladores de nodos de sensores inalámbricos especificando la estructura y funcionamiento para un buen uso.
- Comparar los simuladores de sensores inalámbricos de código abierto en base a diferentes métricas para su aplicación en redes de sensores inalámbricos.

2. Aspectos metodológicos

2.1 Materiales

2.1.1 Recursos Bibliográficos

Este trabajo monográfico se realizó con los siguientes recursos bibliográficos:

- Publicaciones informáticas.
- Libros, revistas y demás materiales informáticos.
- Biblioteca digital de la Universidad Agraria.
- Bibliotecas de otras Universidades.
- Portales web de libros digitales.

2.1.2. Materiales e insumos

A continuación, los siguientes materiales e insumos que se utilizaron:

- Computador portátil o de escritorio.
- Dispositivos de almacenamiento.
- Microsoft office y otras herramientas de tratamiento de texto.
- Impresora.
- Papelería y otros materiales de oficina.

2.1.3 Recursos humanos

Se contó con la asistencia del Tutor monográfico que trabajó con el alumno desarrollador, sumando la colaboración de otros docentes de la Universidad Agraria del Ecuador.

2.2 Métodos

2.2.1 Modalidad y tipo de investigación

El tipo de investigación, orienta la finalidad general del estudio y la manera como se recogieron los datos. Para seleccionar el tipo de investigación se

consideró el propósito, las estrategias, los objetivos y otros aspectos en los que se encuentran inmersos la investigación.

Este trabajo se enmarca dentro del tipo de investigación descriptiva, lo que permitió medir la información recolectada para luego describir, analizar e interpretar sistemáticamente las características del tema con base en la realidad del escenario planteado.

2.2.2 Tipos de métodos

El enfoque metodológico a usar se enmarca en la investigación bibliográfica, documental y de tipo monográfico, basándose en la búsqueda, compilación, análisis los simuladores de red de sensores inalámbricos, por medio de artículos, notas, libros, ensayos y otras fuentes de información como el Centro de Información de la Universidad Agraria del Ecuador y su biblioteca virtual. A continuación se detallan los diferentes métodos utilizados en este trabajo monográfico.

2.2.2.1 Método bibliográfico

Este método es importante ya que permitió realizar la investigación bibliográfica y documental, mediante el uso de libros, folletos, revistas y demás fuentes de datos informáticos relacionados a los simuladores de red de sensores inalámbricos, lo que permitirá recopilar los datos como características, ventajas, entre otros, que luego serán sujetos al análisis.

2.2.2.2 Método inductivo

Con esta metodología, y apoyado por la observación, se partió de lo particular indicando las características de los simuladores de red de sensores inalámbricos para concluir de forma general con una comparación de estos simuladores.

2.2.2.3 Método deductivo

Dentro del proceso de la investigación este método permitió un análisis general de las características de los simuladores de red de sensores inalámbricos. Todos los análisis se derivaron en aspectos particulares que generaron conocimientos que aportaron una mayor comprensión de cada uno de ellos. El método deductivo trata de diseminar aspectos generales para llegar a premisas particulares.

2.2.2.4 Método analítico

Este método distinguió elementos que permitieron descomponer los procesos múltiples y variados, en partes o definiciones básicas en lo que respecta al análisis de los simuladores de red de sensores inalámbricos. Este análisis ayudó a una comprensión específica de las características específicas para poder realizar una comparación.

2.2.3 Técnicas

Para la realización del presente trabajo monográfico, se utilizó la técnica de investigación bibliográfica, permitiendo la recopilación, el análisis y la selección de la información, donde se fundamenta y complementa la investigación con las teorías de diferentes autores.

2.3 Marco legal

Esta investigación monográfica se basó en los siguientes artículos

Constitución de la República del Ecuador:

Sección octava

Ciencia, tecnología, innovación y saberes ancestrales

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad:

1. Generar, adaptar y difundir conocimientos científicos y tecnológicos.
2. Recuperar, fortalecer y potenciar los saberes ancestrales.

3. Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir.

Art. 386.- El sistema comprenderá programas, políticas, recursos, acciones, e incorporará a instituciones del Estado, universidades y escuelas politécnicas, institutos de investigación públicos y particulares, empresas públicas y privadas, organismos no gubernamentales y personas naturales o jurídicas, en tanto realizan actividades de investigación, desarrollo tecnológico, innovación y aquellas ligadas a los saberes ancestrales. El Estado, a través del organismo competente, coordinará el sistema, establecerá los objetivos y políticas, de conformidad con el Plan Nacional de Desarrollo, con la participación de los actores que lo conforman.

Art. 387.- Será responsabilidad del Estado:

1. Facilitar e impulsar la incorporación a la sociedad del conocimiento para alcanzar los objetivos del régimen de desarrollo.
2. Promover la generación y producción de conocimiento, fomentar la investigación científica y tecnológica, y potenciar los saberes ancestrales, para así contribuir a la realización del buen vivir, al sumak kawsay.
3. Asegurar la difusión y el acceso a los conocimientos científicos y tecnológicos, el usufructo de sus descubrimientos y hallazgos en el marco de lo establecido en la Constitución y la Ley.
4. Garantizar la libertad de creación e investigación en el marco del respeto a la ética, la naturaleza, el ambiente, y el rescate de los conocimientos ancestrales.
5. Reconocer la condición de investigador de acuerdo con la Ley.

Art. 388.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento. Un porcentaje de estos recursos se destinará a financiar proyectos mediante fondos concursables. Las organizaciones que reciban fondos públicos estarán sujetas a la rendición de cuentas y al control estatal respectivo. (Asamblea Constituyente, 2018).

3. Análisis y revisión de la literatura

3.1 Simuladores de sensores inalámbricos orientados a la red

3.1.1 Simulador de sensores inalámbricos Castalia

Castalia es un simulador para redes de sensores inalámbricos para comprobar algoritmos y protocolos básicos. Fue construido por el programa (Networks and Pervasive Computing) de National ICT Australia. Este simulador está basado en la plataforma OMNet++ que puede ser utilizada por investigadores y desarrolladores que necesitan probar el comportamiento del algoritmo, del protocolo de una manera real utilizando canales inalámbricos y modelos de radio frecuencia también con un comportamiento caso real de los nodos (Chio, Tibaduiza, Aparicio, y Caro, 2011).

Castalia es un simulador de código abierto utilizado para evaluar redes inalámbricas de sensores. Esta herramienta se ejecuta como módulo adicional sobre el simulador OMNET++ (Guerrero, Alvarez, García, Mayoral, y Contreras, 2017)

Castalia desarrollada por Networks and Pervasive Computing de National, ICT Australia; J-SIM es útil tanto para simular como emular una red de sensores inalámbricos (Blandón y López, 2011).

Castalia contiene un modelo de canal de transmisión avanzado que define las pérdidas en la ruta de transmisión, así como las variaciones en las intensidades de la señal que se producen durante la transmisión. La movilidad del sensor también se ha implementado. El modelo utilizado para el canal de radio se basa en mediciones realistas (Muñoz, 2018, pág. 30).

De acuerdo a lo expuesto por los autores Castalia es un popular simulador WSN basado en el marco OMNet ++, que proporciona un análisis de primer orden

de algoritmos y protocolos antes de la implementación real en una plataforma de nodo específica. El modelado detallado de radio y comunicación es una característica importante de este simulador. Incluye un modelo de radio física, ajustable y altamente preciso junto con un modelo de canal de comunicación avanzado basado en datos medidos empíricamente. Para manejar la entrega de paquetes, Castalia proporciona un cálculo dinámico de interferencia de señal de radio y relación de ruido, en función del tipo de modulación, la intensidad de la señal y otros parámetros.

La estructura de Castalia se refleja en la jerarquía de directorios en el código fuente (archivos.cc.h) que define su comportamiento, y en el lenguaje NED (archivos.ned) que define a los módulos, por ejemplo, nombre, parámetros, puertas de entrada y salida del mismo (Boulis, 2011).

Castalia es uno de los simuladores más realistas, en relación con el canal inalámbrico, frente a otros simuladores como NS2, NiXIM, OPNET, Manassim. Castalia es un simulador ajustable, donde muchos parámetros pueden ser reconfigurados para simular un entorno real; sin embargo, la creación de un modelo de canal inalámbrico es bastante complejo si se tiene en cuenta un entorno cambiante, como por ejemplo una WBAN (Bergamini, Crociani, y Vitaletti, 2009).

Los autores exponen que el comportamiento del simulador Castalia es definido a través de la jerarquía de directorios; y los módulos los define el lenguaje NED, como son nombre, parámetros, entre otros. Este simulador es considerado el más realista, siendo ajustable y pudiendo reconfigurarse los parámetros para simular un entorno real; no obstante es compleja la creación de un modelo de canal inalámbrico.

El reciente interés por las WBANs y el desarrollo de su estándar IEEE 802.15.6 ha creado un impulso importante para que investigadores y desarrolladores de NICTA, establezcan bancos de pruebas experimentales para capturar miles de mediciones. Estas mediciones se analizan para proporcionar modelos precisos tanto para las pérdidas de trayectoria en todo el cuerpo como para el comportamiento de la variación temporal del canal (Averos, 2017).

Castalia es un simulador de redes de sensores inalámbricos - WSN y de redes de área corporal – BAN, que incluye un modelo de canal inalámbrico avanzado, modelos de radio, modelos de sincronización del reloj de nodo, modelos de consumo de energía, varios protocolos MAC, incluyendo IEEE 802.15.4 y protocolos de enrutamiento (Nicta, 2013).

En Castalia los nodos se conectan a través del módulo de canal inalámbrico y además están vinculados a través de los procesos físicos que controlan. Cada nodo es la composición de un módulo de comunicación, un gestor de sensor, un módulo de aplicación, un administrador de recursos y un gestor de movilidad. El módulo de comunicación consiste en tres capas: la radio (capa física), la capa MAC y la capa de enrutamiento, y proporciona capacidades de comunicación al módulo de aplicación. El gestor de movilidad es responsable de la ubicación del sensor en el área. El gestor de recursos modela el uso de los recursos locales, tales como el consumo de energía, el uso de la memoria y los estados de CPU (Nastasi y Cavallaro, 2011).

En castalia hay disponibles tres modelos de interferencias:

1. interferencia = 0, es decir, no hay colisiones aunque varios nodos estén transmitiendo al mismo tiempo, sin embargo, hay algo de aleatoriedad en el módulo de radio modelado como ruido térmico (probabilidad de recepción

de paquetes se basa en SNR), por lo que los paquetes recibidos desde un nodo no son exactamente iguales en dos pruebas.

2. interferencia = 1, modelo simple basado en el concepto de rango de interferencia, en el cual dos transmisiones parcialmente solapadas son ambas descartadas.
3. interferencia = 2, modelo de interferencia aditiva, en el cual la relación de interferencia de la señal se calcula teniendo en cuenta todas las posibles interferencias de otros nodos de sensores. Para todos los escenarios, los efectos de sombreado han sido inhabilitados y se utiliza la aplicación Through put Test, todos los nodos envían paquetes a un nodo coordinador a una velocidad constante (Gómez, Posada, y Vallejo, 2014).

Según lo expuesto por los autores Castalia proporciona un cálculo dinámico de interferencia de señal de radio y relación de ruido en función del tipo de modulación, la intensidad de la señal y otros parámetros. El modelo PHY, a su vez, incluye múltiples estados con retrasos de transición entre ellos, diferentes niveles de transmisión y modulaciones. Además, Castalia permite el modelado de unidades de red móvil; incluyendo modelos personalizables de los protocolos de enrutamiento y MAC más populares para aplicaciones WSN. También proporciona modelos de sensores integrados, que incluyen efectos tales como detección de ruido y sesgo del dispositivo. Castalia ofrece modelado consciente de la energía e informes de consumo de energía para componentes de radio. Además, puede admitir el modelado simple de reloj para CPU. Este simulador está bien documentado y proporciona ejemplos de diseño para facilitar su adopción por parte del usuario final.

3.1.2 Simulador de sensores inalámbricos MiXim

MIXIM es un framework de modelado OMNeT++ para redes inalámbricas móviles y fijas como las redes inalámbricas de sensores, redes ad-hoc, redes vehiculares, etc. Ofrece modelos detallados de la propagación de ondas de radio, la estimación de la interferencia, el consumo de energía de radio transmisor y los protocolos MAC, por ejemplo Zigbee, protocolos de alto nivel de comunicación inalámbrica (Bedoya, 2013).

MiXiM es un marco de referencia diseñado principalmente para el modelado de redes inalámbricas móviles como lo son las redes de sensores, redes ad-hoc y redes vehiculares. Este marco de referencia es una suite de modelos entre los que se incluyen: modelos de propagación de canal, interferencia, consumo de energía, capa física, capa MAC, movilidad, entre otros (Mixim, 2015).

Dentro de los modelos que componen a MiXiM, se cuenta con un modelo MAC de IEEE 802.15.4. Este modelo cuenta con las siguientes propiedades:

- Opera en modo sin paquetes fijo.
- Soporta el modo de operación con y sin acuses de recibo.
- Implementa el algoritmo CSMA/CA como una máquina de estados finita.
- Implementa dos tarjetas de red con las cuales fue validado en una cama de pruebas: TI CC1100 y TI CC2420.
- Puede trabajar modularmente con protocolos de movilidad y de capas superiores que estén adaptados para trabajar dentro del marco de referencia de MiXiM.
- Se encuentra basado en las especificaciones del estándar IEEE 802.15.4 en su versión del 2006.

- No cuenta con las rutinas implementadas para realizar el proceso de asociación y disociación (Reyes, 2017).

El marco de simulación MiXiM se centra en la movilidad de nodos y modelos de canales precisos. MiXiM en sí es una combinación de tres marcos de simulación. El Marco de movilidad proporciona la estructura general, la movilidad y la gestión de conexiones para el marco. Los modelos de propagación de radio se incluyen desde el simulador de canales. Los modelos disponibles en Positif Framework, MAC Simulator y Mobility Framework se utilizan como bibliotecas de protocolos (Mostl & Springer, 2015).

MiXiM, un simulador mixto que combina varios marcos de simulación desarrollados para simulaciones inalámbricas y móviles en OMNeT ++; proporciona modelos y protocolos detallados, así como una infraestructura de soporte. Los componentes básicos de MiXiM se pueden dividir en cinco grupos:

- **Modelos de entorno:** En una simulación, solo se deben reflejar las partes relevantes del mundo real, como los obstáculos que dificultan la comunicación inalámbrica.
- **Conectividad y movilidad:** Cuando los nodos se mueven, su influencia en otros nodos en la red varía. El simulador tiene que rastrear estos cambios y proporcionar una representación gráfica adecuada.
- **Recepción y colisión:** En las simulaciones inalámbricas, los movimientos de objetos y nodos influyen en la recepción de un mensaje. El manejo de la recepción es responsable de modelar cómo una señal transmitida cambia en su camino hacia los receptores, teniendo en cuenta las transmisiones de otros remitentes.

- **Apoyo al experimento:** El apoyo a la experimentación es necesario para ayudar al investigador a comparar los resultados con un estado ideal, ayudarlo a encontrar una plantilla adecuada para su implementación y apoyar diferentes métodos de evaluación.
- **Biblioteca de protocolos:** La biblioteca de protocolos permite a los investigadores comparar sus ideas con las ya implementadas.

De acuerdo a lo manifestado por los autores MiXiM es un marco modular que combina varios simuladores OMNeT ++. Uno de ellos es el Marco de movilidad (MF), los otros son un simulador MAC, un simulador para localización local - Positif y un simulador dirigido a modelos de propagación de radio - ChSim. Con un objetivo diferente, estos cuatro simuladores se ejecutaban en OMNeT ++ y por tener mucho en común se unieron para formar un nuevo simulador llamado MiXiM. Este simulador proporciona soluciones combinando los enfoques de varios marcos de simulación existentes en uno: el soporte de movilidad, la gestión de conexiones y la estructura general se toma del Marco de movilidad (MF); los modelos de radio propagación se toman del simulador de canal (ChSim); y la biblioteca de protocolos se toma del simulador MAC, el marco Positif y del Marco de movilidad. La biblioteca de protocolos MAC abarca desde CSMA simple hasta IEEE802.11.

En la Figura 2 se puede ver una descripción general de una simulación MiXiM, que combina múltiples frecuencias de radio y objetos que influyen en la propagación de radio.

3.1.3 Simulador de sensores inalámbricos Pawis

Dentro del proyecto PAWiS se desarrolla un marco de simulación para Wireless Sensor Networks (WSN). Proporciona funcionalidad para simular los nodos de

la red con su estructura interna, así como la red entre los nodos. Una característica principal es la simulación contemporánea del consumo de energía de cada nodo. El marco se basa en el simulador de eventos discretos OMNeT ++. El modelo definido por el usuario (expresado en clases de C ++) se compila en un simulador ejecutable (Pawis, 2019).

El marco PAWiS que se centra especialmente en la simulación precisa del consumo de energía de una red. Con este simulador es posible establecer una red eléctrica compuesta por proveedores y consumidores de energía, que se influyen mutuamente. Además, proporciona una separación clara de los componentes de software y hardware y permite el modelado de hardware arbitrario, como un transceptor de radio, CPU o una interfaz de sensor (Mostl & Springer, 2015).

El marco de simulación de Pawis se basa en el marco de simulación de red OMNET ++. Para simular y optimizar nodos WSN, el nodo se divide en módulos que se modelan por separado, un pre módulo de clase C ++. Estos módulos se comunican a través de las llamadas interfaces funcionales, que son básicamente llamadas a procedimientos remotos e implementadas como métodos C ++. Cada módulo puede ofrecer e implementar múltiples interfaces funcionales, que pueden invocar interfaces funcionales de otros módulos. Para proporcionar esta funcionalidad, las clases de OMNET ++ están envueltas en clases de Pawis, que luego son la clase principal para los módulos (Haase & Grimm, 2015).

Pawis es otro entorno de diseño dedicado a WSN. Al igual que sensorSim, tiene en cuenta la arquitectura interna del nodo (módulos), así como el entorno que comprende red, perturbaciones y entorno físico y permite la inclusión de

fenómenos WSN específicos, como el análisis del consumo a diferentes niveles de granularidad, soporte de movilidad, fenómenos dinámicos - entorno y canal y aspectos temporales como la latencia. Interfaces estándar y una herramienta de análisis de potencia completan la herramienta. Se diferencia del marco de movilidad por un tratamiento diferente del canal de comunicación y una herramienta de análisis de consumo de energía (Khan, Khan, & Ali, 2016).

Según lo expuesto por los autores el marco de simulación PAWiS facilita el diseño y la simulación de modelos de red de sensores inalámbricos. El enfoque principal se da a la eficiencia energética y, por lo tanto, a capturar las ineficiencias en varios aspectos del sistema. Estos aspectos incluyen todas las capas del sistema de comunicación, la clase de aplicación específica en sí, la fuente de alimentación y la administración de energía, la unidad central de procesamiento (CPU) y la interfaz sensor-actuador. Este marco de simulación se basa en el simulador de eventos discretos OMNeT ++ y proporciona características específicas para simular, analizar y optimizar los aspectos antes mencionados, ofreciendo la implementación de módulos como hardware, software o módulos mixtos.

3.1.4 Simulador de sensores inalámbricos WSNet

WSNet es un simulador controlado por eventos para redes inalámbricas. Sus características principales:

- Simulación de nodos: los nodos simulados se construyen como un conjunto arbitrario de bloques que representan un componente de hardware, un componente de software o un comportamiento / recurso del nodo.
- Simulación del entorno: WSNet ofrece la oportunidad de simular fenómenos físicos (por ejemplo, fuego) y medidas físicas (por ejemplo,

temperatura, humedad). Los nodos pueden leer estos valores, evolucionar en el tiempo y los fenómenos físicos pueden afectar a los nodos, como destruirlos.

- Simulación de medio de radio: WSNNet ha sido diseñado para ofrecer una amplia gama de modelado de medio de radio, desde una capa física ideal básica sin interferencia, sin pérdida de trayectoria y un rango de radio fijo hasta uno complejo y preciso con una fórmula de propagación de Friis, Rayleigh desvanecimiento, frecuencias múltiples y propiedades de correlación complejas entre frecuencias, modulación bpsk, patrones complejos de radiación de antena, entre otros.
- Extensibilidad: Nodo, entorno y bloques de medio radio se desarrollan en bibliotecas dinámicas independientes. En consecuencia, la adición de nuevos modelos no requiere modificar el núcleo de WSNNet y se puede hacer fácilmente (Citi lab, 2019).

WSNet implementa modelos para radio 2.4GHz CC2420 y protocolo CSMA-CA sin ranura 802.15.4, con la posibilidad de seleccionar entre bandas de 800 MHz, 900 MHz y 2400 MHz. Además implementa la modulación O-QPSK dentro de su modelo de radio 802.15.4 de 2.4 GHz (Stelsko, Smolka, & Matyas, 2012).

WSNet es un simulador de red inalámbrica modular controlado por eventos, su arquitectura consiste en diferentes bloques que modelan características y propiedades del medio radio y los nodos sensores. La escalabilidad de WSNNet, muestra la aceleración de la simulación (tiempo simulado sobre tiempo de simulación) para un número variable de nodos y un uso variable del medio de radio. Aunque la aceleración disminuye hasta 0.001 para 1000 nodos y un uso

de ancho de banda de 1 KB / s, el simulador sigue siendo eficiente en cuanto a su precisión. Además, es muy raro tener topologías de red donde el vecindario de comunicación es tan alto como 1000 nodos y en las topologías más convencionales con, por ejemplo, una vecindad de 50 nodos, la velocidad sigue siendo alta (Fraboulet, Chelius, & Fleury, 2016).

WSNet es un framework de código abierto impulsado por eventos que utiliza el lenguaje de programación C++, compatible con las extensiones WSN, pero en cuanto a las extensiones WMSN solamente XS y WSNet (Ali, 2016).

WSNet es un simulador de red inalámbrica modular controlado por eventos. Su arquitectura consta de diferentes bloques que modelan las características y propiedades del medio de radio. La lista de protocolos MAC disponibles también es relativamente rica y contiene el protocolo inalámbrico IEEE 802.15.4. Durante una simulación, el comportamiento de un bloque se especifica utilizando un modelo que es una implementación particular de las funcionalidades del bloque. Los modelos se proporcionan con el simulador o los desarrollan los usuarios (Tampouratzis, Dollas, Brokalakis, & Nikitakis, 2016).

De acuerdo a lo manifestado por los autores WSNet es un simulador para redes de sensores inalámbricos utilizados para la simulación de nodos, de entorno, de medio radio y extensibilidad. La simulación de medio de radio proporciona un modelado de canal de radio realista apropiado para probar la comunicación inalámbrica en DTN móvil. Está diseñado con una escalabilidad extrema como objetivo principal. Primero, se distribuyen los nodos de red de sensores inalámbricos simulados en una variedad de máquinas reales que se comunican a través de una red cableada. Una simulación a gran escala es

emulada por varias simulaciones a menor escala que se ejecutan de manera simultánea y colaborativa.

3.2 Simuladores - emuladores de sensores inalámbricos

3.2.1 Simulador - emulador Tossim

Tossim es un simulador de eventos discretos a nivel de bit para WSNs que operan bajo el sistema operativo TinyOS. Utiliza como lenguaje de programación nesC, lenguaje derivado de C. Una de las limitaciones que se encuentran en este simulador es que no soporta obtener información sobre el consumo energético de los diferentes nodos; otra limitación es que a medida que el tráfico de la red aumenta, el rendimiento decrece considerablemente debido a que el simulador opera a nivel de bit. A pesar de esa limitación, TOSSIM puede realizar simulaciones de WSNs con más de 1000 nodos (Campo, 2015).

Es un simulador de eventos discretos que está incluido en TinyOS. Simula la ejecución de código nesC en nodos Mica, permitiendo la emulación de nodos reales mediante el mapeado de las interrupciones hardware en eventos discretos. Los componentes de emulación de hardware son compilados junto con los componentes reales de TinyOS, utilizando el compilador de nesC. De esta forma, se obtiene un programa ejecutable real de TinyOS que es ejecutado sobre una capa física simulada (Esparza, 2017).

Se trata de un simulador basado en eventos discretos para aplicaciones de WSN basadas en TinyOS. El objetivo es que los usuarios puedan probar una aplicación para en TOSSIM antes de cargar el programa en los nodos. Esto permite a los usuarios depurar, probar y analizar los algoritmos en un entorno

controlado y repetible. El objetivo principal de TOSSIM es proporcionar una simulación fidedigna de las funcionalidades de TinyOS (Roselló, 2009).

De acuerdo a lo expuesto por los autores TOSSIM es un simulador de eventos discretos que puede simular miles de nodos que ejecutan aplicaciones completas. Al reemplazar solo unos pocos sistemas TinyOS de bajo nivel, puede capturar el comportamiento de la mota, lo que permite una amplia gama de experimentación. Aprovechar varias características de las redes de sensores TinyOS permite lograr esta fidelidad a escala. Como las motas individuales tienen recursos limitados de almacenamiento y CPU, TOSSIM puede simular muchas de ellas a la vez. La ejecución basada en eventos de TinyOS se adapta bien a una simulación de eventos discreta, que requiere un motor de simulación muy simple.

Desde la perspectiva de Tossim una porción de código se ejecuta de manera instantánea, mientras que la velocidad del reloj es un valor en torno a 4 MHz. Esto provoca que algunos eventos no ocurran hasta que el código se complete en lugar de ocurrir en el preciso instante. No modela el mundo real, pero permite abstracciones de este. El hecho de que Tossim capture el funcionamiento a bajo nivel puede provocar que programas que no funcionen en la vida real, si funcionen en las simulaciones. Por ejemplo una interrupción muy larga puede provocar que el nodo real se rompa mientras que en el simulador funcionará sin ningún problema (Gordón y Vásquez, 2012).

TOSSIM es un simulador y emulador de redes inalámbricas de sensores, simulando el código TinyOS para los Nodos MICA; que ofrece ventajas evidentes en aquellos proyectos que utilizarán los nodos mencionados, convirtiéndose en el simulador específico para la plataforma. Compila el código TinyOS, y el ejecutable obtenido puede correrse en una PC común. Posee una

GUI, TinyViz, donde se puede observar e interactuar con las simulaciones en curso (Sosa, 2011).

De acuerdo a expresado por los autores las herramientas de simulación como TOSSIM no son realistas. Con TOSSIM es difícil detectar fallas que pueden ocurrir en configuraciones reales, por ejemplo, fallas que ocurren debido a tiempos, enlaces de comunicación, sobrecarga de cálculo de tareas y error de configuración de longitud de mensaje. Además, no admite la estimación del consumo de energía de los nodos sensores. Para mejorar su utilidad para los desarrolladores de TinyOS, TOSSIM tiene mecanismos que permiten a las GUI proporcionar visualización y actuación detalladas de una simulación en ejecución

Toda la simulación se ejecuta en un único hilo, lo que conduciría a una escalabilidad media si se tiene en cuenta que es un simulador orientado a eventos. No obstante, se advierte que el modelo de interferencia puede tener un costo de 10MB de memoria por nodo simulado. Esto reduciría seriamente la escalabilidad (Marelli, 2015).

El modelo de radio de TOSSIM consiste en definir individualmente ganancia y conexiones para cada nodo y en utilizar un modelo de interferencia para determinar si un mensaje es entregado o no. El modelo de interferencia se denominada CPM (closest-pattern matching) y se basa en el análisis estadístico de información de ruido obtenida en forma empírica (Lee, Cerpa, y Levis, 2007).

El simulador está implementado como módulos en lenguaje C pero se incluyen clases C++ que empaquetan (wrappers) estos módulos y proporcionan el framework para utilizar el simulador. A partir de estas clases se pueden generar módulos para Python. Como crítica a este esquema, debe mencionarse que

TOSSIM no es un simulador completo; termina siendo un framework y para utilizarlo es necesario realizar un programa que se encargue de instanciar e inicializar los nodos y de ejecutar el bucle principal de la simulación. Esto conduce a una usabilidad baja (TinyOS, 2013)

Según lo indicado por los autores la simulación de red en TOSSIM está limitada por la homogeneidad del modelo de ejecución. Solo puede ejecutar el mismo programa en cada nodo de la red. Además, el modelo de simulación TOSSIM no es extensible y solo admite un único modelo de plataforma de hardware Micaz. La arquitectura de simulación TOSSIM proporciona un alto nivel de escalabilidad y velocidad de ejecución para redes con una gran cantidad de nodos de sensores. Sin embargo, el modelo de hardware abstracto disponible en TOSSIM no captura detalles de bajo nivel de temporización e interrupciones, lo que puede ser importante para un análisis preciso de potencia de tiempo.

3.2.2 Simulador - emulador Cooja

Cooja es un simulador flexible diseñado para simular redes de sensores corriendo sobre Contiki OS. Con esta herramienta se simulan todas las WSN con sus diferentes configuraciones que se van a diseñar, previo al paso de probarlas en un entorno real. Para la ejecución se utiliza la máquina virtual Instant Contiki 2,7 que proveen los creadores de ContikiOS, que se abre utilizando el entorno de virtualización VMware (Benavente, 2018).

Cooja Network Simulator junto al emulador MSPSim permite desarrollar y verificar aquellas redes que por su gran tamaño resulten difíciles de testear físicamente. Este simulador hace que esa tarea resulte bastante sencilla ya que proporciona un entorno de simulación que permite emular incluso dispositivos hardware reales en una interfaz gráfica sencilla de utilizar (Vera, 2017).

Para la implementación y la simulación de las aplicaciones se utiliza el Sistema Operativo Contiki y su herramienta de simulación Cooja. Desde el propio sitio web de Contiki OS se ofrece ya preparada y de modo gratuito una máquina virtual con el sistema operativo Contiki disponible y el simulador Cooja configurado [#]. La máquina virtual es de tipo VMWare, software que también se encuentra disponible de modo gratuito. Como no se considera una implantación real de la aplicación, no es necesario hardware adicional (Ibarzabal, 2014).

Según lo dispuesto por los autores Cooja es un simulador desarrollado con el lenguaje Java, que se encuentra incluido en el sistema operativo Contiki y creado principalmente para redes inalámbricas; se encuentra compuesto por sensores que ejecutan este sistema operativo en particular. Cooja también está disponible como parte de Instant Contiki, una máquina virtual que contiene un entorno de desarrollo de Contiki listo para usar. Cooja permite simular grandes y pequeñas redes de nodos Contiki. Los nodos se pueden emular a nivel de hardware, que es más lento pero permite una inspección precisa del comportamiento del sistema, o en un nivel menos detallado, que es más rápido y permite la simulación de redes más grandes.

COOJA simula nodos de RDS23 donde cada nodo puede ser de diferente tipo, no solo en el software del nodo sino también en el hardware de la simulación. COOJA es flexible en muchos de sus aspectos, se puede extender fácilmente su funcionalidad introduciendo modificaciones. Un nodo simulado en COOJA tiene tres propiedades básicas:

- Memoria de datos
- Tipo de nodo

- Hardware periférico (Asensio, Miguel, & Pascual, 2009)

Su uso es relativamente sencillo y existen gran cantidad de manuales para poder iniciarse en él. En la interfaz de usuario se puede apreciar una serie de plug-in importantes.

1. Network: Básicamente muestra la disposición de las motas en un plano de dos dimensiones. Útil si queremos extrapolar la distancia a la realidad y saber en qué rangos nos podemos mover en aplicaciones reales de motas.
2. Simulation Control: Manejo principal de la simulación, des de aquí podemos empezar, pausar o parar la simulación. También existe la opción que nos permite lanzar otra vez los proyectos a las motas: muy útil si hemos hecho modificaciones en el código.
3. Mote output: Escucha los supuestos puertos series de las motas. Podemos ver la salida estándar o StdOutput (Linux).
4. Timeline: Información temporal de los eventos producidos de cada mota. Estado de la radio entre otras características (Carbonell, 2015).

Otra premisa que se debe cumplir es que COOJA debe conocer las direcciones de memoria de cada una de las variables y funciones de cada sistema ContikiOS que se carga. Esto se realiza a través del mapa de archivos que se genera en la biblioteca. Con ello, COOJA puede averiguar las direcciones de cualquier variable que se localice en la citada biblioteca compilada. Gracias a lo anterior, el simulador es capaz de ver o modificar determinadas variables, en función de las cuales, podrá hacer una u otra acción, que es, básicamente, el modo de funcionamiento de un plugin (Fraga, 2015).

Según lo expuesto por el autor el uso del simulador de redes de sensores inalámbricos Cooja es sencillo de usar, existiendo manuales disponibles que

facilitan su uso; la simulación la realiza con nodos RDS23 que pueden ser de diversos tipos; esta herramienta es flexible, pudiendo extender su funcionalidad introduciendo modificaciones. Entre las propiedades del simulador Cooja están memoria de datos, tipo de nodo, hardware periférico. Además en la interfaz de usuario se observan plug-in como network, simulation control, mote output y timeline.

3.2.3 Simulador - emulador Avrora

Avrora es un simulador y emulador de tipo GPL, con capacidad para generar código que luego puede portarse a la familia MOTE de sensores inalámbricos. Se trata, por tanto, de soluciones hechas pensando en una solución tecnológica muy concreta (López, Ruíz, & Marcelín, 2009)

Es un proyecto de investigación del UCLA Compilers Group, es un conjunto de herramientas de simulación y análisis de programas escritos para microcontroladores AVR y los nodos de WSN, Mica2 (Roselló, 2011). Avrora es un simulador escrito en JAVA y de código abierto, para ejecutar programas de simulación integrados en TinyOS. Permite simular sobre dos plataformas Mica2 y MicaZ. Este proyecto de investigación nació del grupo de Compiladores UCLA, que buscan herramientas y soluciones para el micro controlador AVR de Atmel, ofreciendo un api en Java para realizar las simulaciones requeridas (Ramos & Velandia, 2014).

Avrora es una herramienta de simulación que permite simular ejecuciones de programas en microcontroladores con precisión de ciclo de reloj. Sin embargo, solo permite simular dos tipos muy específicos de plataforma HW (Díaz, 2017).

Según lo expresado por los autores Avrora, fue desarrollado como un proyecto de investigación del Grupo de Compiladores de UCLA, es un simulador de código abierto con precisión de ciclo para programas de detección integrados. A diferencia de otros simuladores, que pueden simular solo plataformas específicas, Avrora tiene independencia del lenguaje y del sistema operativo. Proporciona un marco para el análisis de programas, lo que permite la comprobación estática de software embebido y una infraestructura para futuras investigaciones de análisis de programas. Avrora simula una red de motas, ejecuta los programas de microcontroladores reales, en lugar de modelos del software, y ejecuta simulaciones precisas de los dispositivos y la comunicación por radio.

Avrora está formado por un conjunto de herramientas diseñadas para la simulación y análisis de programas implementados para los microcontroladores de Atmel (Atmega 128, Atmega 32 y Atmega16), que son los microcontroladores que usan las motes de la familia Mica e Iris, por lo que Avrora se puede usar con programas diseñados para cualquiera de ellas. El lenguaje de programación en el que Avrora está desarrollado es Java, por lo que es fácil de ejecutar en casi cualquier entorno, siendo sólo necesaria la instalación de la máquina virtual de Java JVM. Avrora dispone de un conjunto de monitores que permiten analizar los recursos de la aplicación simulada, tales como energía consumida por los componentes, números de ciclos de CPU usados por cada uno de ellos o el tiempo en el que cada componente ha estado en un determinado estado. También se puede analizar la memoria consumida por la aplicación, simular varias aplicaciones a la vez sobre la misma red de sensores inalámbrica, incluso simular la aplicación en tiempo real (Arias, 2015).

El simulador de redes inalámbricas de sensores Avrora es una herramienta que estima con precisión de ciclo de reloj la ejecución de programas que se implementan en nodos de redes de sensores inalámbricas (Díaz, Avrora simulador para redes de sensores inalámbricos, 2014).

Avrora proporciona puntos de instrumentación flexibles para insertar modelos de radio más complejos que la simple recepción de bytes completos, al tiempo que conserva la precisión de temporización. Por ejemplo, se puede usar un modelo de radio más complejo para simular la pérdida parcial del preámbulo. Esto sucede porque aunque el software envía y recibe bytes individuales de la radio CC1000, una radio receptora puede no recibir los primeros bits de una transmisión, ya que intenta bloquear la señal, lo que hace que los bytes posteriores entregados al software se desplacen por algún número de bits (Marelli, 2015).

De acuerdo a lo indicado por los autores Avrora es un emulador y un simulador de nivel de código, que se utiliza para emular el hardware del sensor o para procesar el código del programa como lo estaría en un dispositivo de hardware real. Esta herramienta es un marco de línea de comandos compatible con las plataformas de sensores MEMSIC Mica2 y MicaZ, lo que permite la simulación de redes inalámbricas, la instrumentación dinámica y el análisis estático; además simula y analiza para programas escritos para microcontroladores AVR. La característica especial de Avrora es que funciona en el nivel de instrucción, es decir, los programas de microcontroladores reales se pueden ejecutar en el simulador, en lugar de solo simular modelos de software.

3.3 Comparación de los simuladores de red de sensores inalámbricos de código abierto

Mediante la combinación de modelos de canal y radio inalámbricos conocidos, los desarrolladores de Nicta, encontraron que la mayoría de los simuladores disponibles para WSN (NS2, MiXim, OPNET, Manassim, TOSSIM) estaban por debajo de la situación actual en modelos ideales de canal inalámbrico (sin pérdidas). Por ende en la simulación de la comunicación, el impacto sobre el resultado puede ser significativo; los modelos siguen siendo simplistas o no aptos para comunicaciones de corto alcance de baja potencia, a pesar de la existencia de modelos adecuados que se ha desarrollado en los últimos años, razón principal para que investigadores de NICTA decidieran desarrollar un propio simulador (Averos, 2017).

No todos los simuladores de redes de sensores inalámbricos ejecutan al mismo tiempo las siguientes actividades: ejecutar código TinyOS, ejecutar códigos diferentes en distintos nodos, ver los paquetes enviados, realizar filtrado de los paquetes y ver un grafo con los enlaces de comunicación utilizados durante la simulación. Los simuladores que cumplen con la mayoría de esos requerimientos son Cooja y Avrora; sin embargo, Cooja no permite filtrar los paquetes capturados, por lo que en el grafo de la topología se muestran tanto los paquetes de datos como los de enrutamiento. Dado que los paquetes de enrutamiento se envían por broadcast en el grafo resultante no se puede observar cuál es realmente la estructura de la red y cuál es el padre elegido por cada nodo. Por otro lado, el simulador Avrora no tiene interfaz gráfica ni filtrado de paquetes. En las simulaciones se muestra un dump del contenido de las tramas que fueron enviadas. Dada esta limitante para realizar las pruebas

sobre los ataques a los protocolos de capa de red implementados, se decidió integrar la herramienta que se iba a implementar con el simulador Avrora (Segovia, 2014).

A continuación se comparan las características de las herramientas de simulación WSN de código abierto más recientes en sus últimas versiones, como son Castalia, MiXim, Pawis, WSNet, Tossim, Cooja, Avrora, basado en un conjunto de criterios descritos anteriormente.

La tabla 1 proporciona un resumen de los simuladores orientados a la red que se han detallado anteriormente, donde se evidencia que los simuladores Castalia, Mixim y Pawis se basan en OMNet ++, mientras que WSNet usa un motor de simulación personalizado. El lenguaje de programación que utilizan los simuladores Castalia, Mixim, Pawis y WSNet es C++, este último también emplea Python. En cuanto a la portabilidad/plataforma en su mayoría emplean Linux, a excepción de WSNet que emplea Posix. Solamente los simuladores Mixim y Pawis, pueden admitir una Gui.

La tabla 2 suministra un resumen de los simuladores / emuladores orientados a la red que se han detallado anteriormente, donde se evidencia que Tossim, Cooja y Avrora se basan en TinyOS, no obstante Cooja también utiliza Contiki. El lenguaje de programación que utiliza Tossim es C++; Cooja y Avrora manejan Java. En cuanto a la portabilidad de la plataforma en su mayoría emplean Linux. Solamente Cooja, pueden admitir una Gui.

4. Conclusiones

En base a lo expuesto en el presente trabajo monográfico se presentan las siguientes conclusiones:

La simulación es una herramienta esencial para estudiar las redes inalámbricas de sensores debido a la inviabilidad del análisis y las dificultades de establecer experimentos reales. Los simuladores de redes inalámbricas de sensores (WSN) son una creación rápida de prototipo, como apoyo para el desarrollo, pruebas y evaluación de este tipo de sistemas integrados. Entre los simuladores que más se utilizan se encuentran: Castalia, MiXim, Pawis y WSNNet. Los simuladores WSNNet y Castalia son herramientas muy eficientes para modelado de red a gran escala, mientras que Mixim y Pawis requieren una mayor cantidad de recursos de computación y tiempo para ejecutar las mismas configuraciones de red.

También existen simuladores – emuladores que cumplen ambas dos funciones, es decir que el emulador de red permite que dispositivos reales interactúen a través de la red simulada; entre los más utilizados se encuentran: Cooja y Avrora, que están desarrollados con el lenguaje Java; y, Tossim, que emplea lenguaje C++ y Python. Los simuladores/emuladores Avrora y Cooja son eficientes en el modelado de red a gran escala, mientras que TOSSIM requiere una mayor cantidad de recursos de computación y tiempo para ejecutar las mismas configuraciones de red.

La mayoría de los simuladores comparados soportan el modelado del consumo de energía y el modelado de la movilidad, y son extensibles. Los simuladores Castalia, Mixim, Pawis, WSNNet, Cooja y Avrora, pueden modelar el consumo de energía, por contra TOSSIM que no puede modelar el consumo de energía. En cuanto a la movilidad de modelado, los simuladores Castalia, Mixim, Pawis,

WSNet, Cooja y Avrora, admiten movilidad, no así TOSSIM que no pueden modelar la movilidad. Con respecto al número de nodos admitidos por cada simulador, Castalia, Tossim, Mixim, Pawis y WSNet, pueden simular una red compuesta por 1000 nodos. Para una red grande, podemos usar Avrora y Cooja que pueden admitir una gran cantidad de nodos hasta 10000 nodos. Los simuladores Castalia, Pawis, WSNet, Tosim, Cooja y Avrora son extensibles. Todas las herramientas presentan resultados similares en términos de rendimiento de red y modelado de comunicación para varios tamaños de red y velocidades de paquetes.

5. Recomendaciones

De acuerdo con el análisis realizado en el trabajo de investigación se realizan las siguientes recomendaciones:

El uso de los simuladores WSNNet y Castalia para modelado de red a gran escala.

El uso de Avrora y Cooja como simuladores/ emuladores por ser excelentes herramientas para el modelado de red a gran escala.

Realizar pruebas comparativas en escenarios reales.

6. Bibliografía

- Ali, H. (2016). Simulación de redes inalámbricas de sensores multimedia (WMSN). *Revista de Publicació de ciencias e investigación*, 2(3), 51-66.
- Arias, M. (2015). Avrora. *Módulo de análisis del tráfico y energía de aplicaciones de redes de sensores simuladas con Avrora*, 37. Madrid, España: Universidad Carlos III de Madrid.
- Asamblea Constituyente. (2018). Constitución del Ecuador. *Registro oficial No. 449 Reforma*. Quito, Ecuador.
- Asensio, A., Miguel, A., & Pascual, J. (2009). Simulador para redes inalámbricas de sensores Cooja. *Diseño de un simulador para redes de sensores*, 41. Madrid, España: Universidad Complutense de Madrid.
- Averos, J. (2017). Simuladores Castalia. *Estudio del estándar IEEE 802.15.6 y simulación de los parámetros de transmisión en una red de área corporal en la banda de frecuencia de 2,4 GHz*, 131-133. Quito, Ecuador: Escuela Politécnica Nacional.
- Bedoya, A. (2013). Simulador de sensores inalámbricos MiXim. *Estudio de entornos de simulación en redes de vehículos*, 66. Zaragoza, España: Universidad de Zaragoza.
- Benavente, C. (2018). Simulador de redes inalámbricas Cooja. *Diseño, simulación y despliegue de redes inalámbricas de sensores de ingeniería y sistemas de telecomunicación*, 13. Madrid, España: Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación.
- Bergamini, L., Crociani, C., & Vitaletti, A. (2009). Simulador Castalia. *Simulación vs bancos de pruebas reales: una validación de simuladores WSN*, 18. Roma, Italia: Universidad de Roma.

- Blandón, J., & López, J. (2011). Bio Route: Unasimulador para redes de sensores inalámbricos. *Revista Educación en Ingeniería*, 7(13), 23-31.
- Boulis, A. (2011). Castalia. Un simulador para redes de sensores inalámbricos y redes de área corporal. *NICTA, Tecnología de información y comunicación*, 5-10.
- Campo, C. (2015). Simulador Tossim. *Uso de una red inalámbrica de sensores para estimar riesgo de incendio*, 34. Madrid, España: Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación.
- Carbonell, E. (2015). Simulador de redes inalámbrica de sensores Cooja. *Desarrollo de una red de sensores inalámbricos para la monitorización de la actividad física*, 14. Catalunya, España: Universidad Politécnica de Catalunya.
- Chio, N., Tibaduiza, D., Aparicio, L., & Caro, L. (2011). Redes de sensores inalámbricos. *Congreso Internacional de Ingeniería Mecatrónica* (págs. 4-19). Bucaramanga: Revista Universidad Autónoma de Bucaramanga.
- Citi lab. (2019). *Un simulador basado en eventos para redes de sensores inalámbricos a gran escala (WSNet)*. Obtenido de <http://www.citi-lab.fr/portfolio/wsnet-simulator/>
- Díaz, A. (2014). Avrora simulador para redes de sensores inalámbricos. *Plataforma virtual para el análisis del rendimiento y la seguridad en redes de sensores inalámbricas*, 9. Santander, España: Universidad de Cantabria.
- Díaz, A. (2017). Avrora simulador de redes inalámbricas de sensores. *Sistemas embebidos en red seguros*, 30. Santander, España: Universidad de Cantabria.

- Eriksson, J., Osterlind, F., & Kopke, A. (2010). Hacia simulaciones comparables de objetos cooperantes y redes inalámbricas de sensores. *Primer taller internacional ICST sobre metodología y herramientas de rendimiento para redes inalámbricas de sensores* (págs. 978-963). Pisa: ICST.
- Esparza, J. (2017). Simulador Tossim. *Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos y enrutamiento para obtener el menor costo energético en redes zigbee*, 38. Guayaquil, Ecuador: Universidad Católica de Santiago de Guayaquil.
- Fraboulet, A., Chelius, G., & Fleury, E. (2016). World sens: Herramientas de desarrollo y creación de prototipos para redes de sensores inalámbricos de aplicación específica. *15° Simposio Internacional sobre procesamiento de información en redes de sensores* (págs. 176-185). Vienna: IEEE.
- Fraga, A. (2015). Simuladores de redes de sensores inalámbricos Cooja. *Simulador Cooja para WSN basado en el sistema operativo Contiki*, 51. Santa Clara, Cuba: Universidad Central Marta Abreu de Las Villas.
- Glaser, J., Weber, D., Madani, S., & Mählknecht, S. (2012). Marco de simulación Power Aware para redes y nodos de sensores inalámbricos. *Revista sobre sistemas embebidos*, 1, 1-16.
- Gómez, P., Posada, G., & Vallejo, M. (2014). Evaluación del desempeño del protocolo de enrutamiento AODV para diferentes escenarios de redes de sensores inalámbricos. *Ingeniería y Desarrollo. Revista de la división de ingenierías de la Universidad del Norte*, 8-16.
- Gordón, M., & Vásquez, B. (2012). Simuladores Tossim. *Análisis de desempeño de redes de sensores inalámbricos en tiempo real aplicadas a*

- monitorización volcánica*, 63. Salgolquí, Ecuador: Escuela Politécnica del Ejército.
- Guerrero, J., Alvarez, O., García, L., Mayoral, M., & Contreras, J. (2017). Simulación de una WSN aplicada en entornos marinos. *Revista Telemática*, 16(3), 99-107.
- Haase, J., & Grimm, C. (2015). Métodos para el diseño de nodos de red de sensores inalámbricos de ultra baja potencia. En G. Nicolescu, I. O'Connor, & C. Piquet, *Tecnología de diseño para sistemas embebidos heterogéneos* (págs. 460-461). Lituania: Springer.
- Ibarzabal, E. (2014). Simuladores de red de sensores inalámbricos Cooja. *Aplicación para una red WSN basada en ContikiOS*, 17. Bilbao, País Vasco: Universidad del País Vasco.
- Khan, S., Khan, A., & Ali, N. (2016). Simulación centrada en la energía y exploración espacial de diseño para WSN. En *Redes de sensores inalámbricos. Estados actual y tendencias futuras* (págs. 215-248). Florida, USA: CRC Press.
- Lee, H., Cerpa, A., & Levis, P. (2007). Mejora de la simulación inalámbrica a través del modelado de ruido. *Actas del VI Simposio internacional sobre procesamiento de información en redes de sensores* (págs. 21-30). Cambridge: IEEE.
- López, M., Ruíz, M., & Marcelín, R. (2009). YAWSS otro simulador para redes inalámbricas de sensores. *VII Congreso internacional en innovación y desarrollo tecnológico* (págs. 402-409). Cuernavaca: CIINDET.
- Marelli, R. (2015). Avrora, simulador de redes de sensores inalámbricas. *DEVSTOSSIM: un frameworks DEVS para simular redes de sensores*

- inalámbricos que utilizan el sistema operativo TinyOS*. Buenos Aires, Argentina: Universidad de Buenos Aires.
- Marelli, R. (2015). Simulador Tossim. *DEVS-TOSSIM: Un frameworks DEVS para simular redes de sensores inalámbricos que utilizan el sistema operativo TinyOS*, 45. Buenos Aires, Argentina: Universidad de Buenos Aires.
- Mixim. (2015). *Mixim project (mixed simulator)*. Obtenido de <http://mixim.sourceforge.net>
- Mostl, G., & Springer, A. (2015). Integración de Mixim, Pawis y Steam-Sim: Combinación de modelos de canales, coincidencia energética y código de aplicación de la vida real. *Actas de la OMNET ++ Community Summit*, (págs. 4-8). USA.
- Muñoz, V. (2018). Castalia. *Análisis del modelado de redes de sensores inalámbricos mediante la validación de la plataforma de simulación OMNET*, 30. Guayaquil, Ecuador: Universidad Católica de Santiago de Guayaquil.
- Nastasi, C., & Cavallaro, A. (2011). WISE-MNet: un entorno experimental para redes inalámbricas de sensores multimedia. *Conferencia procedimientos de señales de sensores para defensa* (págs. 1-5). Londres: SSPD.
- Nicta. (2013). *Castalia framework*. Obtenido de Nicta: <http://castalia.research.nicta.com.au/index.php/en/>
- Omnet. (2020). *Manual de Instalación Castalia*. Obtenido de Omnet: <http://omnet-manual.com/castalia-installation/>
- Pawis. (2019). *Pawis simulación Framework*. Obtenido de <http://pawis.sourceforge.net/>

- Ramos, P., & Velandia, D. (2014). Avrora simulador de redes de sensores inalámbricos. *Simulación de una red de acelerómetros para el monitoreo del estado de la malla vial en un sector de la ciudad de Bogotá, un enfoque de aplicación de Its e IOT*, 48-49. Bogotá DC, Colombia: Universidad Católica de Colombia.
- Reyes, L. (2017). Simulador de red de sensor inalámbrica Mixim. *Análisis del impacto del número de usuarios y tasa de datos ofrecida en el traspaso entre resumideros de un WBAN/WPAN enfocada a aplicaciones de sistemas del cuidado de la salud*, 48. Baja California, México: Centro de investigación científica y de educación superior de ensenada, Baja California.
- Roselló, V. (2009). Tossim. *Modelado para simulación de redes de sensores inalámbricas predespliegue basado en visualsense*, 21. Madrid, España: Universidad Politécnica de Madrid.
- Roselló, V. (2011). Simulador de redes inalámbricas Avrora. *Modelado para simulación de redes de sensores inalámbricas predespliegue basado en visualsense*, 23. Madrid, España: Universidad Politécnica de Madrid.
- Segovia, M. (2014). Simuladores de redes de sensores inalámbricos cooja y avrora. *Seguridad en redes de sensores inalámbricos*, 27-28. Montevideo, Uruguay: Universidad de la República.
- Sosa, E. (2011). Simulador Tossim. *Contribuciones al establecimiento de una red global de sensores inalámbricos interconectada*, 152. La Plata, Argentina: Universidad Nacional de La Plata.
- Stelsko, A., Smolka, T., & Matyas, V. (2012). WSNET, un simulador basado en eventos para redes de sensores inalámbricos a gran escala. 5°

- Conferencia Internacional sobre herramientas y técnicas de simulación* (págs. 75-84). Desenzano del Garda: Simutools.
- Tampouratzis, N., Dollas, A., Brokalakis, A., & Nikitakis, A. (2016). D2.1: Primera implementación del simulador de procesamiento. *WP2 Subsistema de procesamiento y simulación de red*, 12. Italia: Cossim.
- TinyOS. (2013). *Tossim*. Obtenido de TinyOS: <http://docs.tinyos.net/tinywiki/index.php/TOSSIM>
- Vera, J. (2017). Simulador de redes inalámbricas de sensores Cooja. *Análisis e implementación de redes inalámbricas de sensores deterministas robustos*, 42. Valencia, España: Universidad Politécnica de Valencia.

7. Glosario

Enrutamiento: También conocido como encaminamiento, se trata de la función de buscar un camino entre todos los posibles caminos elegibles en una red, en base a alguna métrica.

Escalabilidad: es un término usado en tecnología para referirse a la propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer el funcionamiento y calidad normales del mismo. Cuando un sistema tiene esta propiedad, se le refiere comúnmente como “sistema escalable” o, simplemente, escalable

Gateway: Dispositivo o pasarela que permite la interconexión entre dos interfaces de red.

Java: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

Mote: Se denomina mote (o nodo sensor) al dispositivo con sensores y capacidad para realizar mediciones y transmitirlos, que colabora con otros para formar una WSN.

nesC: Es un dialecto del lenguaje de programación C optimizado para las limitaciones de memoria de los nodos sensores.

Sensor: Transductor o dispositivo cuyo fin es recibir estímulos y transformarlos en información, permitiendo así medir fenómenos físicos.

Simulador: Dispositivo o aparato que emula un fenómeno, el funcionamiento real de otro aparato o dispositivo o las condiciones de entorno a las que están sometidos una máquina, aparato o material.

Topología: Es la forma en que está desplegada la red, sea en el plano físico o lógico.

TinyOS: Es un sistema operativo de código abierto basado en componentes para redes de sensores inalámbricas.

8. Anexos

Tabla 1. Comparación de los simuladores Castalia, Mixim, Pawis, WSNnet, Tossim, Avrora y Cooja

	Simuladores			
	Castalia	Mixim	Pawis	WSNet
Motor de simulación	OMNET ++	OMNET ++	OMNET ++	Motor de simulación personalizado
Lenguaje de programación	C ++	C ++	C ++	C++, XML
Escalabilidad en tiempo de ejecución de simulación > 100 nodos	Alto / 1000	Bajo / < 100	Medio / 500	Alto /1000
Modelo de canal de comunicación	Modelo experimental de pérdida de ruta SINR	Modelo de pérdida, espacio libre, registro, sombreado	No	Espacio libre, registro. Shadowing, Rayleigh fading
Nodos móviles	Si	Si	Si	Si
Plataformas HW compatibles	Plataforma virtual	Plataforma virtual	Plataforma virtual	Plataforma virtual
Modelo para el componente transceptor de radio	cc 2420, cc 1100	cc 2420, cc 1100	cc 2420	cc 2420, cc 1100
Protocolos de nivel MAC	Tmac, IEEE 802.15.4	IEEE 802.15.4, BMAC, IEEE 802.11	Simple CSMA, IEEE 802.11, CSMA-MPS	IEEE 802.11

Se describe la comparación de los simuladores Castalia, Mixim, Pawis y WSNnet

(Baquerizo, 2020)

Tabla 2. Comparación simuladores/emuladores Tossim, Avrora y Cooja

	Simuladores / emuladores		
	Tossim	Avrora	Cooja
Motor de simulación	Motor de simulación personalizado	Motor de simulación personalizado en Java	Motor de simulación personalizado en Java
Lenguaje de programación	C++/Python,	Java, AVR	Java - Contiki
Escalabilidad en tiempo de ejecución de simulación > 100 nodos	Alto	Alto	Medio
Número de nodos	1000	10000	500
Modelo de canal de comunicación	Espacio libre con gráfico dirigido de errores de bit	Espacio libre	Espacio libre, Iniciar sesión. Shadowing
Nodos móviles	No	No	No
Plataformas HW compatibles	MicaZ	MicaZ, Mica2	Tmote sky
Modelo para el componente transceptor de radio	cc 1100	cc 2420	cc 2420
Modelado de energía	de Plataforma MicaZ	Mica 2	Para cada módulo HW
Herramientas para estimación de batería / vida útil de nodos	No/no	No/no	No/no
Precisión de tiempo / potencia	Alto/alto	Alto/alto	Alto/alto

Se describe la comparación de simuladores/emuladores Tossim, Avrora y Cooja

(Baquerizo, 2020)

	Cooja	Avrora	TOSSIM
Sistema operativo	Contiky	TinyOS	TinyOS
Nivel de simulación	Multinivel	Instrucción	Sistema operativo
Modelo de ejecución	Dirigido por el tiempo	Dirigido por eventos	Dirigido por eventos
Distintos tipos de nodo	Si	Diferente software	No
Interfaz de usuario	Interfaz gráfica	Línea de comandos, interfaz gráfica limitada	No, es un framework
Modelos de radio	Basado en ecuación de Friis o definición de probabilidad de recepción exitosa	Basado en ecuación de Friis, no simula interferencia	Ganancia y conexiones para cada nodo, modelo de interferencia basado en análisis estadístico

Figura 1. Comparación simuladores - emuladores
(Marelli, 2015)

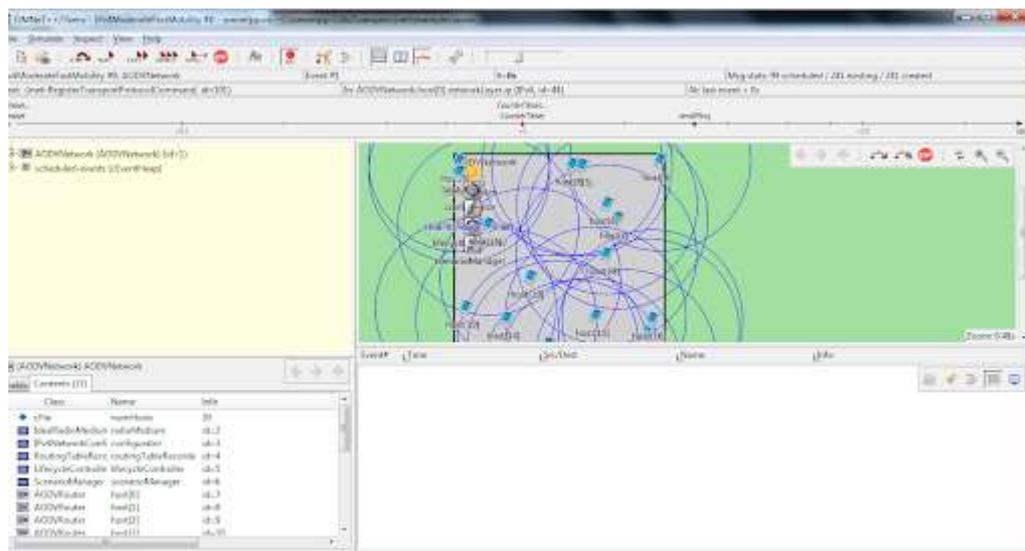


Figura 2. Instalación de Simulador Castalia
(Omnet, 2020)

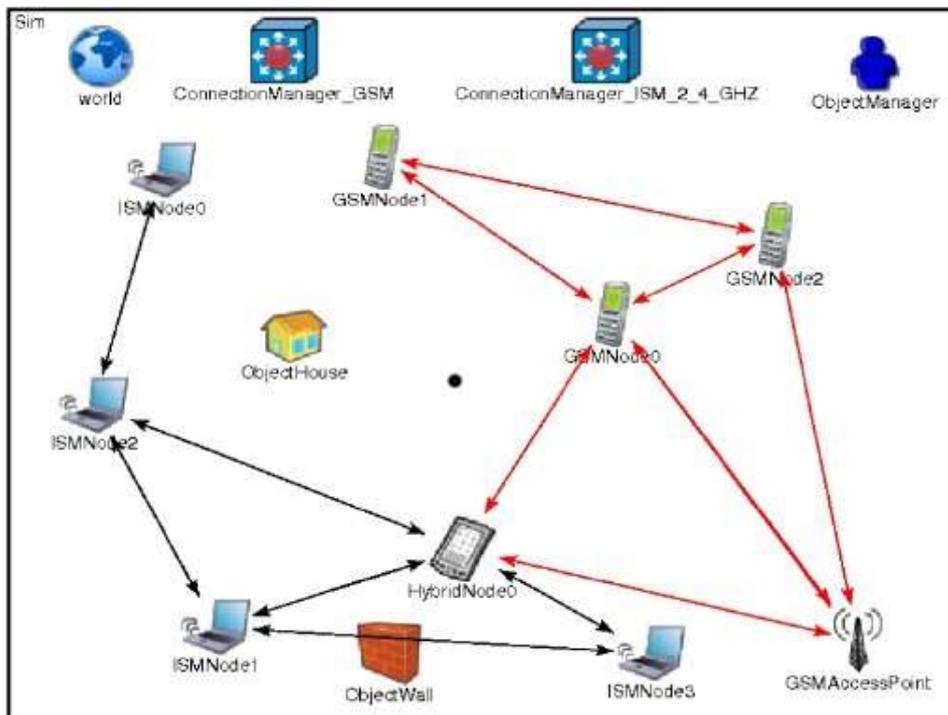


Figura 3. Simulador Maxim

(Eriksson, Osterlind, & Kopke, 2010)

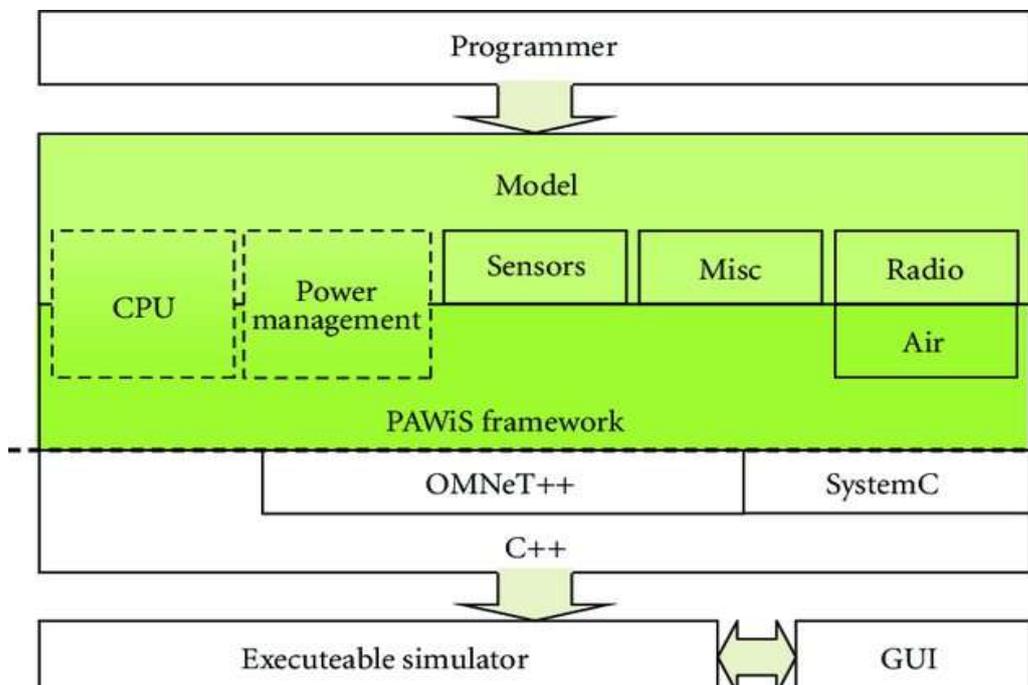


Figura 4. Estructura del marco de simulación PAWiS

(Glaser, Weber, Madani, & Mahlkecht, 2012)

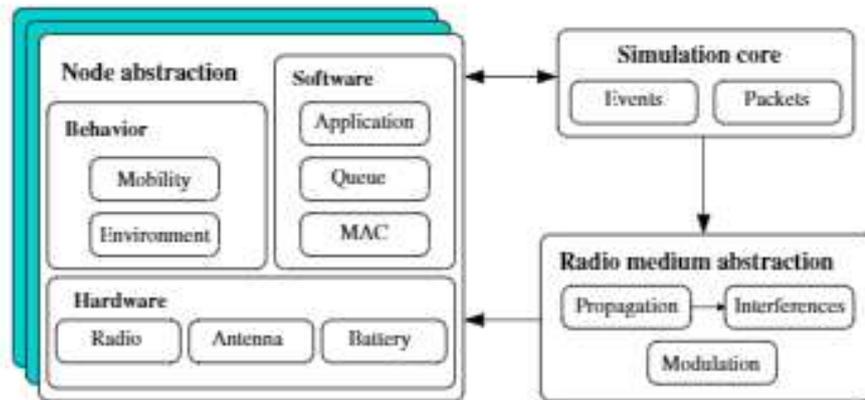


Figura 5. Arquitectura de bloques de WSNet

(Fraboulet, Chelius, & Fleury, 2016)

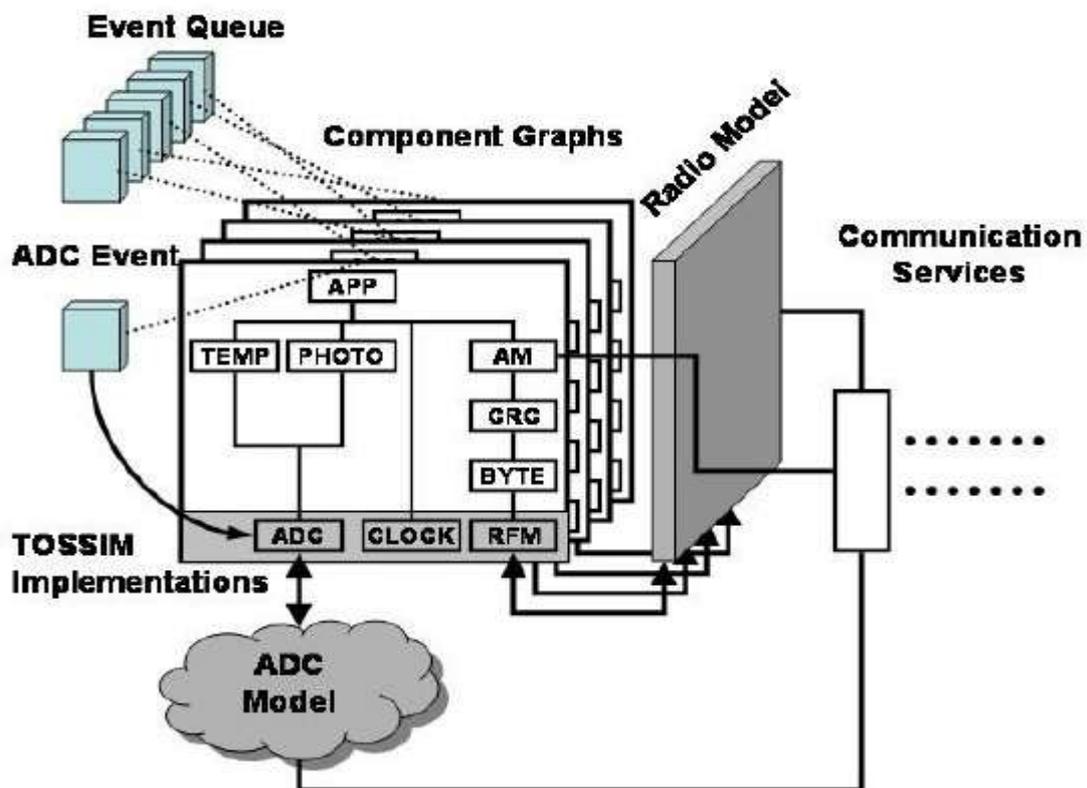


Figura 6. Arquitectura TOSSIM: marcos, eventos, modelos, componentes y

servicios

(Nicta, 2013)



Figura 7. Ventana principal del simulador Cooja

(Fraga, 2015)

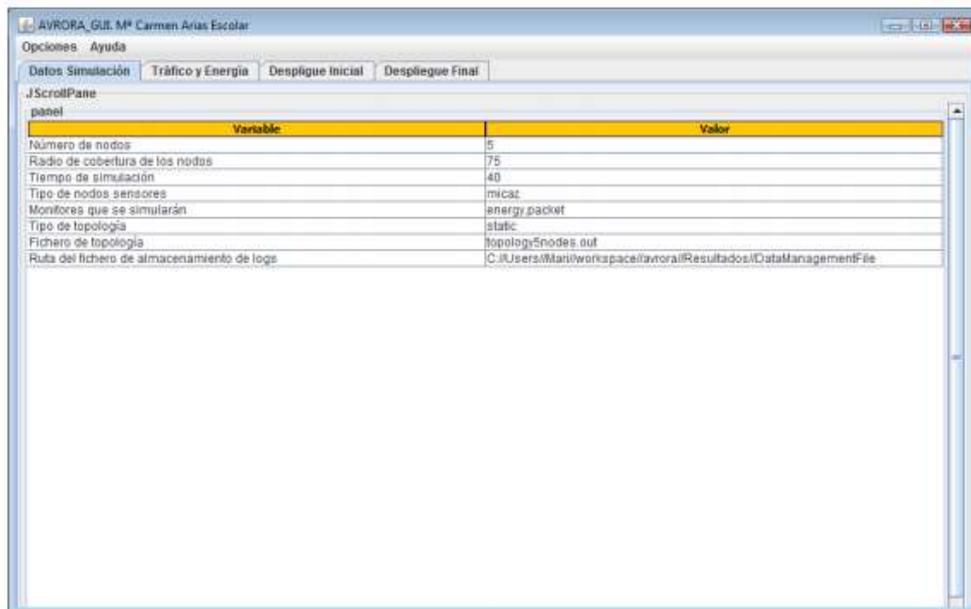


Figura 8. Ventana/pantalla datos de simulación - Castalia

(Arias, 2015)