Universidad Politécnica de Valencia

Departamento de Informática de Sistemas y Computadores

Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos

Tesis Doctoral presentada por:
Carlos Lino Ramírez

Dirigida por:
Carlos Tavares Calafate
Arnoldo Díaz Ramírez

Marzo 2012

Dedicada a

El amor de mi vida Kary

por enseñarme que la felicidad existe.

A Derek

por brindarme la dicha de ser padre.

A mi madre

por darme la vida y estar siempre a mi lado.

Agradecimientos

Un gran reto sin duda alguna concluir un doctorado, un gran reto que se logra gracias a la ayuda y apoyo directo e indirecto de muchas personas.

Dentro de las primeras personas a quien quiero agradecer su ayuda invaluable por haberme guiado y apoyado en todo momento se encuentra mi asesor de tesis Dr. Carlos Calafate, una persona con mucho talento y mucha sabiduría quien desde el inicio del proyecto de tesis me ofreció su apoyo para lograr terminar con éxito este trabajo. También quiero expresar mi agradecimiento al Dr. Arnoldo Díaz, coasesor de esta tesis doctoral, por su incondicional apoyo y por todas sus sugerencias para lograr terminar la tesis.

Igualmente quiero agradecer a todos los integrantes del Grupo de Redes de Computadores, al director del grupo, Dr. Pietro Manzoni, al Dr. Juan Carlos Cano, y al Dr. Carlos Calafate por permitirme pertenecer al GRC y por brindarme todas las herramientas y lo necesario para desarrollar los trabajos de tesis en el laboratorio de este grupo; muchas gracias por la buena atención que me brindaron a mí y a todos los que pertenecemos al grupo.

Así mismo, hago extensivos estos agradecimientos a todos mis compañeros y amigos del GRC, por su invaluable apoyo en todo momento, por su compañía, por compartir con todos ustedes momentos tan amenos a la hora del café. Muchas gracias a mi gran hermano Johann (por haberme ayudado en todo, tanto en lo profesional como en lo personal), Marga (una persona con mucho talento), Ingrid Juliana (una gran persona), Jorge (muy preparado profesionalmente), Jordi (con varios proyectos y siempre con tiempo para compartir con los demás), Alvaro (con mucho futuro profesional y haciendo la vida mas alegre a todos los del grupo compartiéndonos sus pasatiempos), Pedro (buen compañero con mucho entusiasmo para seguir adelante), y muchos compañeros que tuve la suerte de conocer, aunque fuera por periodos de tiempo cortos, como Peppino, Sascha, Nacho, Jean, Gianluca, Wanes, y Filippo. No puedo dejar de mencionar a mis amigos que, aunque no pertenezcan al grupo GRC, siempre me brindaron su apoyo: Carlino, Diego, Javier, Guillermo, Pepe. También quiero agradecer la invaluable ayuda de mi amigo Luis.

Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos

Carlos Lino Ramírez

Resumen

Las aplicaciones para redes de sensores inalámbricas, o *wireless sensor networks* (WSNs), han mostrado un crecimiento significativo en los últimos años. Actualmente constituyen una alternativa tecnológica interesante para el desarrollo de aplicaciones que requieren monitorizar constantemente el estado de cualquier variable relacionada con escenarios de diversos ámbitos. Si las aplicaciones detectan cambios en los valores de dichas variables, pueden activar la ejecución de acciones preventivas que ayuden a restaurar las condiciones normales del entorno monitorizado.

Algunos ejemplos de aplicaciones que se pueden beneficiar de las WSNs son las aplicaciones para la detección de eventos, entre las que se incluye la detección de incendios forestales. Este tipo de aplicaciones ha recibido mucha atención recientemente, ya que cada año se presentan incendios forestales que arrasan con una gran cantidad de flora y fauna, provocando grandes pérdidas económicas y humanas. Otra área de gran interés es la utilización de redes de sensores en la detección de propagación de gases. Estas aplicaciones tienen la finalidad de evitar tragedias, sobre todo en el caso de la propagación de gases peligrosos. Por otra parte, las redes de sensores también han sido utilizadas en la detección y seguimiento de objetivos e intrusos. Con estas aplicaciones es posible vigilar áreas restringidas, ya sea por el servicio que proporcionan o por los objetos de valor que puedan contener. Como puede observarse, estos tipos de eventos tienen la característica de ser *eventos críticos* donde el tiempo de respuesta del sistema tiene una gran importancia.

Para implementar eficientemente aplicaciones que utilicen redes de sensores inalámbricas en la detección de eventos de propagación de fuego y gas, así como para detectar y realizar el seguimiento de intrusos, es conveniente utilizar mecanismos que permitan detectar los eventos críticos de forma correcta e inmediata, de tal manera que se informe y actúe en tiempo real para llevar a cabo las acciones necesarias. En esta tesis doctoral se propone una arquitectura para redes de sensores que permita detectar en tiempo real la presencia de eventos que alteren el estado normal del entorno monitorizado, actuando a continuación convenientemente. En la arquitectura propuesta se utiliza la tecnología IEEE 802.15.4, y se proponen dos nuevos protocolos de encaminamiento que optimizan el envío de la información a través de las estaciones de la red. Se proponen también algoritmos de agregación de los datos que permiten reconstruir los eventos monitorizados.

El primer protocolo propuesto se denomina *Drain Announcement Based Routing* (DABR), y utiliza un algoritmo de descubrimiento de rutas en el que el dreno o sumidero de datos anuncia su ubicación a todos los nodos que forman la WSN. Con este

protocolo de encaminamiento se pretende reducir la sobrecarga de encaminamiento para el descubrimiento de rutas por los nodos sensores que requieren enviar información al dreno. El algoritmo propuesto permite además reducir el retardo *extremo-a-extremo* al mantener poco tráfico de encaminamiento en los canales de comunicación. Este protocolo está orientado a escenarios en los que los nodos sensores y el nodo dreno son fijos, y están distribuidos en una topología tipo malla.

El segundo protocolo de encaminamiento propuesto en este trabajo es el denominado *Mobile-sink Routing for Large Grids* (MRLG), el cual tiene como principal objetivo reducir el tráfico de control de encaminamiento en escenarios donde el dreno es móvil. Los nodos de la red deberán actualizar su ruta hacia el dreno con la restricción de actualizar únicamente la tabla de rutas de los nodos cercanos al dreno y que hayan sido afectados por su cambio de posición, evitando así la necesidad de modificar la tabla de rutas de los nodos lejanos.

En este trabajo también se proponen algoritmos de agregación de datos que permiten determinar el perímetro afectado en el caso de eventos de gas y fuego, así como la posición de un intruso de una forma dinámica y en tiempo real. Estos algoritmos identifican las zonas en riesgo, ejecutando las acciones necesarias para garantizar la seguridad del área que se desea proteger.

Finalmente, como parte de las herramientas desarrolladas e implementadas para cubrir todos los aspectos del proceso de modelado, se ha desarrollado una plataforma que permite generar y evaluar eventos de propagación interna y externa de gas y fuego, así como patrones de movilidad de intrusos.

Como herramienta metodológica se utilizó el simulador ns-2, el cual ha permitido evaluar los protocolos propuestos bajo el estándar IEEE 802.15.4, analizando el impacto que diferentes parámetros de diseño tienen sobre las prestaciones de los mismos.

Design of a sensor network architecture for supporting applications for event detection applications

Carlos Lino Ramírez

Abstract

The development of applications for wireless sensor networks (WSNs) have grown significantly in recent years. Currently, WSNs are an interesting technological alternative to develop applications that constantly need to monitor the state of any variable in the scope of different types of scenarios. If the applications detect changes in the values of these variables, they can trigger preventive actions that allow restoring the normal conditions in the monitored environment.

Some examples of applications that can benefit from the use of WSNs are event detection applications, among which we have the detection of forest fires. This kind of applications has recently received much attention since, every year, forest fires devastate large areas, damaging flora and fauna, and causing huge material and human losses. Another area of great interest is the detection of gas propagation. The main goal of these applications is to avoid tragedies, especially when dangerous gases are involved. On the other hand, WSNs have also been used in the detection and tracking of targets and intruders. These applications are useful in the surveillance and security of restricted areas where the service offered or the objects therein contained have great value. As we can observe, all these types of events can be classified as *critical events* since the system's response time is of great importance.

To efficiently implement applications that rely on wireless sensor networks for detecting the spreading of fire and gas, as well as for the detection and tracking of intruders, it is convenient the use mechanisms that allow the detection and reporting of critical events to be correct and to be made within a short period, so that the system can inform and act immediately to avoid more serious problems. In this doctoral thesis, an architecture for wireless sensor networks is proposed, that allows the system to do real time detection of events that alter the normal state of the sensed environment, acting in consequence afterwards. The proposed architecture uses IEEE 802.15.4 technology, and two new routing protocols are proposed, which aim to optimize the delivery of information throughout the network. Data aggregation algorithms are also proposed, thereby allowing to reconstruct the monitored events.

The first proposed protocol is the *Drain Announcement Based Routing* (DABR), which uses a route discovery algorithm where the drain announces its location to all sensor nodes that integrate the WSN. This routing protocol aims at reducing the route discovery overhead by sensor nodes attempting to send reports to the drain node. The proposed algorithm also aims at reducing the end-to-end delay by introducing

low routing overhead on the communication channels. This protocol assumes that both the sensor and the drain nodes are fixed (that is, with no mobility), and that the sensor nodes are deployed using a grid topology.

The second proposed routing protocol is the *Mobile-sink Routing for Large Grids* (MRLG), which is intended to reduce the routing control traffic in scenarios where the drain is mobile. The sensor nodes should update their route towards the drain, with the restriction that only those nodes near the drain and affected by its mobility need to update their routing table, thereby avoiding modifying routing tables for those nodes that are far-away.

In this work, new data aggregation algorithms are also proposed, being used to determine the affected area in the case of gas and fire spreading, as well as locating intruders dynamically and in real time. These algorithms identify areas at risk, executing the necessary actions to guarantee the security of the sensed area.

Finally, as part of the tools developed and implemented to cover all aspects of the modeling process, a platform has been developed that allows generating and evaluating both internal and external fire or gas spreading events, as well as intruder mobility patterns.

As a methodological tool we used the ns-2 simulator, which allows evaluating the proposed protocols under the IEEE 802.15.4 standard, analyzing the impact that different design parameters have on their performance.

Disseny d'una arquitectura per a xarxes de sensors amb suport per a aplicacions de detecció d'esdeveniments

Carlos Lino Ramírez

Resum

Les aplicacions per a xarxes de sensors sense fils, o wireless sensor networks (WSN), han mostrat un creixement significatiu en els últims anys. Actualment constitueixen una alternativa tecnològica interessant per al desenvolupament d'aplicacions que requereixen monitorar constantment l'estat de qualsevol variable relacionada amb escenaris de diversos àmbits. Si les aplicacions detecten canvis en els valors d'aquestes variables, poden activar l'execució d'accions preventives que ajuden a restaurar les condicions normals de l'entorn monitorat.

Alguns exemples d'aplicacions que es poden beneficiar de les WSN són les aplicacions per a la detecció d'esdeveniments, entre les quals s'inclou la detecció d'incendis forestals. Aquest tipus d'aplicacions ha rebut molta atenció recentment, ja que cada any es presenten incendis forestals que destrueixen una gran quantitat de flora i de fauna, i provoquen grans pèrdues econòmiques i humanes. Una altra àrea de gran interès és la utilització de xarxes de sensors en la detecció de propagació de gasos. Aquestes aplicacions tenen la finalitat d'evitar tragèdies, sobretot en el cas de la propagació de gasos perillosos. D'altra banda, les xarxes de sensors també han sigut utilitzades en la detecció i el seguiment d'objectius i intrusos. Amb aquestes aplicacions és possible vigilar àrees restringides, siga pel servei que proporcionen o pels objectes de valor que puguen contenir. Com es pot observar, aquests tipus d'esdeveniments tenen la característica de ser esdeveniments crítics on el temps de resposta del sistema té una gran importància.

Per a implementar eficientment aplicacions que utilitzen xarxes de sensors sense fils en la detecció d'esdeveniments de propagació de foc i gas, com també per a detectar i realitzar el seguiment d'intrusos, és convenient utilitzar mecanismes que permeten detectar els esdeveniments crítics de forma correcta i immediata, de tal manera que s'informe i s'actue en temps real per a dur a terme les accions necessàries. En aquesta tesi doctoral es proposa una arquitectura per a xarxes de sensors que permeta detectar en temps real la presència d'esdeveniments que alteren l'estat normal del medi ambient monitorat, per a tot seguit poder actuar convenientment. En l'arquitectura proposada s'utilitza la tecnologia IEEE 802.15.4, i es proposen dos nous protocols d'encaminament que optimitzen l'enviament de la informació a través de les estacions de la xarxa. Es proposen també algorismes d'agregació de les dades que permeten reconstruir els esdeveniments monitorats.

El primer protocol proposat s'anomena drain announcement based routing (DABR), i utilitza un algorisme de descobriment de rutes en el qual el drenatge o l'embornal de dades anuncia la ubicació que té a tots els nodes que formen la WSN. Amb aquest protocol d'encaminament es pretén reduir la sobrecàrrega d'encaminament per al descobriment de rutes pels nodes sensors que requereixen enviar informació al drenatge. L'algorisme proposat permet, a més, reduir el retard extrem-a-extrem en mantenir poc de trànsit d'encaminament en els canals de comunicació. Aquest protocol està orientat a escenaris en els quals els nodes sensors i el node de drenatge són fixos, i estan distribuïts en una topologia tipus malla.

El segon protocol d'encaminament proposat en aquest treball és l'anomenat mobile-sink routing for large grids (MRLG), el qual té com a principal objectiu reduir el trànsit de control d'encaminament en escenaris on el drenatge és mòbil. Els nodes de la xarxa s'han d'actualitzar la ruta cap al drenatge amb la restricció que l'actualització afecta únicament la taula de rutes dels nodes propers al drenatge i que hagen sigut afectats pel seu canvi de posició. Així s'evita la necessitat de modificar la taula de rutes dels nodes llunyans.

En aquest treball també es proposen algorismes d'agregació de dades que permeten determinar el perímetre afectat en el cas d'esdeveniments de gas i foc, i també la posició d'un intrús d'una forma dinàmica i en temps real. Aquests algorismes identifiquen les zones en risc, i executen les accions necessàries per a garantir la seguretat de l'àrea que es vol protegir.

Finalment, com a part de les eines desenvolupades i implementades per a cobrir tots els aspectes del procés de modelatge, s'ha desenvolupat una plataforma que permet generar i avaluar esdeveniments de propagació interna i externa de gas i foc i també patrons de mobilitat d'intrusos.

Com a eina metodològica, s'hi ha utilitzat el simulador ns-2, el qual ha permès avaluar els protocols proposats sota l'estàndard IEEE 802.15.4, tot analitzant l'impacte que diferents paràmetres de disseny tenen sobre les prestacions d'aquests.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Capítulo 1

Introducción

Las redes de sensores inalámbricas (*wireless sensor networks* o WSNs) permiten el desarrollo e implementación de una amplia gama de aplicaciones relacionadas con la monitorización de entornos, fáciles de instalar, fáciles de utilizar, confiables y seguras. En este capítulo se describen brevemente las redes de sensores inalámbricas, y se presentan la motivación y los objetivos que se persiguen en esta tesis doctoral. Finalmente, se indica como se estructura el documento de tesis.

1.1. Redes de Sensores Inalámbricas (WSNs)

En los últimos años el uso de las redes de sensores inalámbricas ha crecido exponencialmente, logrando integrarse en una gran cantidad de áreas de aplicación. Algunas de las razones por las cuales ha logrado este importante crecimiento son el bajo coste de despliegue de los nodos sensores, el bajo consumo de energía, la disponibilidad de nodos sensores multifuncionales, el tamaño reducido de los sensores, y la comunicación a corta distancia entre los mismos.

Una red de sensores inalámbrica es una red cuyos dispositivos principales son nodos que pueden medir las condiciones del ambiente a través de diversos sensores. Estos son dispositivos con la capacidad de recibir y enviar información de forma inalámbrica a uno o mas dispositivos a la vez, tales como ordenadores portátiles, PDAs, teléfonos móviles y, principalmente, pequeños dispositivos equipados con las características básicas para la recepción o monitorización de un evento, el almacenamiento y posterior procesado de la información. En estas redes típicamente se realiza el envío de la información a un dreno o fuente de drenos, los cuales tienen mayor capacidad de procesamiento de la información. Este envío normalmente requiere que la información sea retransmitida por los nodos intermedios, mediante varios *saltos* hasta alcanzar el dreno destino, el cual puede ser fijo o estar en movimiento.

Las características básicas de los dispositivos sensores son: tamaño reducido, bajo consumo de energía, unidad de procesamiento con capacidad limitada, memoria con capacidad para almacenar apenas pequeñas cantidades de información, y capacidad

de comunicarse con otros nodos a una corta distancia; estas características básicas permiten tener un tiempo de vida mayor con respecto a dispositivos con características superiores.

Las redes de sensores inalámbricas pueden ser utilizadas en diferentes áreas, tales como salud, entorno militar, automovilística, ergonomía, aplicaciones industriales, seguridad, aviónica, entornos inteligentes, domótica, agricultura, monitorización, seguimiento de personas o animales y entretenimiento, entre otras. Las redes de sensores inalámbricas han recibido mucha atención recientemente, debido a la gran cantidad de aplicaciones que pueden desarrollarse en este tipo de entornos [1–3].

En el ámbito de este trabajo nos centramos en áreas de aplicación que requieren monitorización de eventos críticos. Dichos eventos ocurren de forma esporádica, pero una vez que se han generado, su monitorización se debe llevar a cabo preferentemente con restricciones de tiempo-real. Un ejemplo de este tipo de aplicaciones lo podemos encontrar en el área de la monitorización de condiciones ambientales. Se pueden monitorizar ambientes internos y externos, supervisar áreas que pueden tener hasta cientos de metros cuadrados y durante varios años. Estas aplicaciones deberán responder, en caso de que se detecte un estado crítico, tan rápido como sea posible. Por lo tanto, estas aplicaciones requieren tecnologías de monitorización con prestaciones de tiempo-real y con alta fiabilidad en el servicio.

Los escenarios que están relacionados con esta área son:

- Monitorización de propagación de incendios forestales.
- Monitorización de propagación de gases tóxicos.
- Detección de inundaciones.
- Monitorización de erupciones volcánicas.

En una WSN los nodos pueden enviar y recibir información de cualquier nodo que se encuentre dentro del rango del radio. Cada nodo sensor puede detectar parámetros del medio ambiente para los que esté preparado, y cada nodo es capaz de comunicar la información detectada a otros nodos o a algún otro dispositivo que esté agregando y procesando la información.

El despliegue de los nodos sensores puede llevarse a cabo según las necesidades del evento que se requiere monitorizar. La cantidad de sensores que forman la red puede ir desde cientos hasta miles de ellos. La posición de cada de uno de los nodos puede seguir un patrón específico o tener una disposición puramente aleatoria. Es importante una elección cuidadosa de la topología de comunicación que se utilizará para encaminar y transmitir la información, de tal manera que se asegure que se alcance al nodo destino (dreno) en el menor tiempo posible y evitando al máximo la pérdida de paquetes de información.

Las WSNs pueden ser completamente autónomas o combinarse con otros tipos de redes, incluso para conectarse a Internet utilizando puntos de acceso inalámbricos. Por otra parte, las WSNs deben adaptarse dinámicamente ante los cambios continuos de las características de la red, tales como la topología, la potencia de la señal, el tráfico de la red y la distribución de la carga. Para soportar algunas de estas características

mencionadas, son importantes los algoritmos y protocolos de encaminamiento, los cuales deberán ser completamente adaptativos, anticipándose al comportamiento futuro de la red a partir de parámetros tales como el nivel de sobrecarga de encaminamiento, el retardo *extremo-a-extremo* y el *throughput* [4].

Además, los recursos de la red deben poder ser localizados y utilizados de forma automática, sin necesidad de una configuración manual previamente establecida. Finalmente, dependiendo de la aplicación, podría ser necesario incorporar técnicas orientadas a ofrecer calidad de servicio (*Quality of Service* o QoS) que permitan ofrecer garantías de servicio sobre determinado tráfico de la red.

En resumen, las principales características de las WSNs son las siguientes:

- Auto-configurables.
- Nodos densamente desplegados.
- Tolerantes a fallos.
- Nodos energéticamente eficientes, debido a sus requerimientos limitados de energía, capacidad de cómputo y memoria.
- Baterías de larga duración.
- Heterogeneidad.
- Se pueden adaptar a una gran cantidad de eventos que se requiera monitorizar.
- Fácil despliegue de los nodos.
- Bajo coste.
- Encaminamiento dinámico y adaptativo.
- Discriminación de distintos tipos de tráfico (QoS).

1.2. Motivación

Debido a los recientes avances tecnológicos en las redes de sensores inalámbricas, es posible desarrollar una gran variedad de aplicaciones basadas en WSNs para casi cualquier área de aplicación, entre las que destacan domótica, medio ambiente, seguridad, salud, entornos industriales y monitorización, entre otros. Sin embargo, existen todavía algunas áreas de aplicación que no han sido abordadas de forma eficiente y en su totalidad, como por ejemplo, la detección y propagación de gas y fuego, y el seguimiento de intrusos. Algunas propuestas orientadas a estas aplicaciones no disponen de protocolos de encaminamiento específicos y, muchas veces, hacen uso de protocolos de encaminamiento para redes *ad hoc* y no para WSNs, ofreciendo prestaciones que no son plenamente eficientes. El estándar IEEE 802.15.4 es el que desde nuestro punto de vista, y para este subconjunto de aplicaciones a estudiar, es el adecuado para utilizarse como base para el desarrollo de nuevos protocolos de encaminamiento y

algoritmos para agregación de datos, ya que el estándar IEEE 802.1.54 está orientado para aplicaciones que requieren de comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de baterías.

De forma general, a continuación se presentan los principales retos de la presente tesis doctoral:

- Combinar la tecnología IEEE 802.15.4 con nuevos protocolos de encaminamiento específicos para datos con requisitos de baja latencia.
- Desarrollar protocolos de encaminamiento especialmente diseñados para la transmisión de información en las WSNs con drenos estático y dinámicos.
- Proponer algoritmos de agregación para determinar el perímetro afectado tanto en eventos de propagación de gas como de fuego, así como estimar la posición de un intruso de forma dinámica y en tiempo real.

1.3. Objetivos

El principal objetivo de este trabajo consiste en diseñar una arquitectura para redes de sensores inalámbricas que considere el uso de dispositivos sensores de bajo coste, y que combine la detección binaria de eventos con la tecnología IEEE 802.15.4. La arquitectura propuesta será utilizada en la monitorización en tiempo real de eventos críticos, tales como la propagación de fuego y gases tóxicos, y el seguimiento de intrusos. Este objetivo general tiene integrados los objetivos particulares, mencionados a continuación:

- Implementar una herramienta modeladora de eventos para WSNs, que incluya propagación de fuego, propagación de gases tóxicos y movimiento de intrusos.
- Diseñar un protocolo de encaminamiento para WSNs basado en la tecnología IEEE 802.15.4, y que sea capaz de llegar a un compromiso en el que al menos se satisfagan los siguientes requerimientos: escalabilidad, mínima sobrecarga de control, tolerancia a fallos y robustez.
- Diseñar un protocolo de encaminamiento, que además de las propiedades del protocolo anterior, tenga un buen rendimiento en escenarios donde el nodo dreno sea móvil. Para lograr este objetivo se utiliza una aproximación que actualice únicamente la tabla de rutas en los nodos cercanos al dreno que experimenten un cambio de topología, manteniendo sin cambio la tabla de rutas de los nodos sensores más alejados, los cuales seguirán conservando rutas válidas.
- Proponer un algoritmo de agregación de datos para entornos de monitorización de fuego/gases en expansión, que permita estimar el borde de dicho evento en tiempo-real, a partir de datos binarios de detección (detectado/no detectado) por parte de cada sensor.

- Evaluar el rendimiento de los protocolos de encaminamiento y los algoritmos desarrollados con ayuda de la herramienta modeladora de eventos, y con el entorno de simulación de redes estándar (ns-2), el cual incluye un modelo detallado del estándar IEEE 802.15.4.

1.4. Estructura de la tesis

La tesis está estructurada de la siguiente manera: en el capítulo 2 se presenta una visión general de las redes de sensores inalámbricas (WSNs) haciendo hincapié en los estándares IEEE 802.15.4 y Zigbee. En el capítulo 3 se describe el modelado y seguimiento de eventos críticos, así como los algoritmos propuestos para la reconstrucción de dichos eventos mediante la agregación de los datos recibidos en el dreno. En el capítulo 4 se desarrollan y analizan las propuestas de encaminamiento de baja sobrecarga en WSNs con drenos estáticos y móviles. El capítulo 5 presenta la evaluación de prestaciones bajo el estándar IEEE 802.15.4 utilizando los protocolos DABR y MRLG, así como resultados de la precisión obtenida en el proceso de reconstrucción de eventos. Por último, en el capítulo 6 se presentan las conclusiones, las publicaciones relacionadas con la tesis y el trabajo futuro.

Capítulo 2

Redes de Sensores Inalámbricas

En este capítulo se presenta una visión general del estado actual de las redes de sensores inalámbricas (WSN), incluyendo su diseño, arquitectura, calidad de servicio y comunicación en tiempo real. Una WSN que incorpora actuadores recibe el nombre de red de sensores y actuadores inalámbrica (*Wireless Sensor and Actuator Network* o WSAN). En este tipo de redes, tanto sensores como actuadores desarrollan tareas específicas y monitorización distribuida. En las WSAN los sensores recolectan información del ambiente físico que, al ser procesada por la estación base, puede generar acciones que son ejecutadas utilizando los actuadores.

2.1. Introducción

La proliferación de dispositivos inalámbricos (teléfonos móviles, tabletas electrónicas, etc) y de aplicaciones de computación ubicua ha provocado que las telecomunicaciones inalámbricas se hayan convertido en una parte muy importante de la vida cotidiana. La tecnología de las comunicaciones ha cambiado increíblemente la forma cómo piensan, se comportan, trabajan y se entretienen las personas.

Los sistemas domóticos pueden utilizar una red cableada o inalámbrica. Los principales sistemas cableados son las líneas telefónicas, *módems* por cable y líneas de transmisión de energía eléctrica. Cada uno de ellos ofrece ventajas y desventajas que están relacionadas principalmente, con el ancho de banda disponible, la instalación, el mantenimiento y el coste, entre otros. La motivación clave para el uso de la tecnología inalámbrica es la reducción de costes por instalación. Las redes inalámbricas permiten lograr altas prestaciones con un mínimo de esfuerzo de instalación. Los sistemas inalámbricos se han impulsado gracias a las nuevas tecnologías, que han logrado un alto grado de integración y un bajo coste de los componentes requeridos.

Dentro del hogar hay muchas aplicaciones que requieren contar con medios de comunicación, tales como Internet, conexión de diversos ordenadores, redes de audio y vídeo, automatización del hogar y seguridad. Cada una de estas aplicaciones tiene diferentes necesidades de ancho de banda, costes y procedimientos de instalación.

Con Internet, la mayor preocupación de los diseñadores es satisfacer la necesidad de compartir conexiones de alta velocidad. Por otro lado, las aplicaciones de automatización del hogar y aplicaciones de seguridad no necesitan esta alta velocidad, ni el uso de protocolos con gran sobrecarga, que afectarían seriamente el consumo de energía, requerirían de mayor poder de procesamiento, y tendrían un alto coste.

Siguiendo con aplicaciones del hogar, si se coloca un detector de temperatura, la temperatura no variará muy rápidamente, por lo que sólo será necesario enviar datos unas pocas veces por hora. Este tipo de aplicaciones funciona muy bien con un enlace inalámbrico de baja potencia y baja transferencia de datos. El uso de cables no es necesario y supondría un alto coste de instalación. Además, los dispositivos consumirían muy poca energía, lo que evitaría el cambio constante de las baterías. La tecnología 802.11 (WLAN) resultaría excesiva y cara para satisfacer los requerimientos de conexión. La tecnología Bluetooth, aunque se concibió originalmente como un sustituto del cable, aún es cara, de poco alcance y alto consumo energético, aunque menor que la WLAN.

En el año 2000, dos grupos especialistas en estándares (ZigBee y el grupo de trabajo IEEE 802) se unieron para desarrollar un nuevo estándar para redes inalámbricas de bajo consumo y de bajo coste, orientado a aplicaciones domóticas e industriales. Como resultado de este trabajo se propuso un nuevo estándar para redes de área personal (LR-WPAN: *Low Rate Wireless Personal Area Network*) que ahora se conoce como el estándar 802.15.4.

Las características más importantes del estándar IEEE 802.15.4 son la flexibilidad de la red, bajo coste y bajo consumo de energía. Este estándar se puede utilizar para muchas aplicaciones, tales como domóticas e industriales, que requieren una baja tasa de transmisión de datos. Esta será la tecnología utilizada en las propuestas de la presente Tesis Doctoral.

2.2. Dispositivos sensores

Los nodos sensores son dispositivos de bajo coste, con capacidades de monitorización, cómputo y comunicación limitados. Por otra parte, los actuadores son nodos equipados con mejores capacidades de procesamiento, transmisores más potentes y con mayor tiempo de vida en la batería.

Los componentes de un nodo sensor se muestran en la figura 2.1. Puede observarse que cuentan con un microprocesador que administra todas las tareas, y con uno o mas sensores, que son los que miden los datos del medio monitorizado. Incluyen también una memoria que se utiliza para almacenar datos temporalmente o durante su procesamiento. También cuentan con un receptor/transmisor de radio con su antena. Todos los dispositivos disponen de una batería que proporciona energía. Normalmente las baterías pueden proporcionar cargas iniciales del orden de 10.000 julios y deberán ser usadas de forma inteligente para tener una duración equivalente al tiempo de vida de la red.

Como resultado de la necesidad de utilizar técnicas que hagan uso eficiente de la energía disponible, las tareas de procesamiento de datos están normalmente dis-

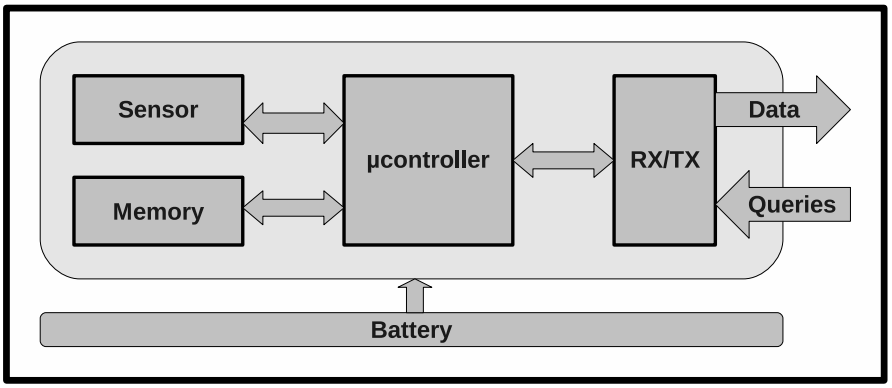


Figura 2.1: Arquitectura de un nodo sensor.

tribuidas en la red. Por lo tanto, los nodos cooperan para enviar los datos hacia el nodo dreno o nodo destino, que normalmente tiene mayores capacidades de memoria, procesamiento y fuente de energía que los nodos sensores.

Para poder explicar el funcionamiento de una red de sensores inalámbrica, primero debe definirse el evento o situación que se desea monitorizar a través de nodos sensores. El evento puede generarse en un espacio abierto o cerrado, y el área cubierta por los sensores será el único espacio que detectará cambios o movimientos en el ambiente. Una vez que se ha definido el área y las medidas del escenario a monitorizar, debe definirse la cantidad de nodos, así como la distribución más adecuada para cubrir por completo el área de interés. Después de seleccionar la topología de distribución de los nodos, y dependiendo del evento a detectar, debe definirse la asociación de los mismos y los grupos de nodos o *clusters* que podrán formarse para monitorizar el evento. Una de las principales características de las redes de sensores inalámbricas es su capacidad para trabajar de manera casi autónoma, sin la necesidad de administración de la red para su buen funcionamiento.

Los nodos sensores desplegados en una zona de actuación, se auto-configuran para producir información de alta calidad acerca del ambiente físico. Cada nodo sensor tiene la capacidad de recolectar y encaminar los datos a otros nodos sensores o al dreno. El dreno puede ser un nodo fijo o móvil capaz de conectar la red de sensores a una infraestructura de comunicación existente o a Internet, de tal manera que el usuario pueda acceder a los datos reportados por la WSN.

La figura 2.2 muestra un escenario que cuenta con 19 nodos sensores y un nodo dreno en una WSN. Un mayor número de sensores permite monitorizar un área geográfica con mayor precisión. Los escenarios con un solo dreno tienen una escalabilidad limitada debido a que, al aumentar la cantidad de nodos fuente, la cantidad de datos enviada hacia el dreno será cada vez mayor y, una vez alcanzada su capacidad máxima de recepción de paquetes, no soportará más información.

Una aproximación a la cantidad de sensores que podrán ser atendidos por el dreno, puede obtenerse de acuerdo a la propuesta de Verdone *et al.* en [5], quienes definen

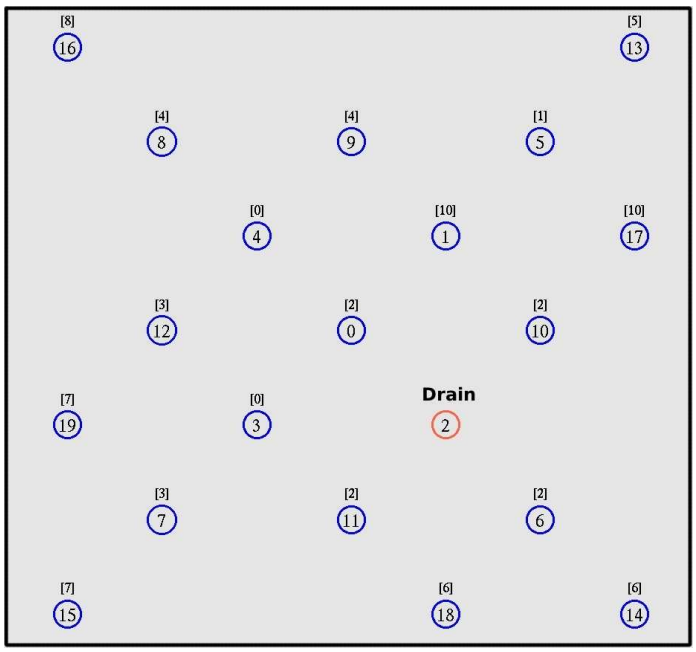


Figura 2.2: Escenario de red de sensores con 20 nodos (19 sensores y 1 dreno).

el número máximo de nodos que pueden conectarse con el dreno. En su propuesta, se asume una WSN en donde los nodos envían paquetes de información de tamaño D bytes cada T_R segundos. Suponen además que los N nodos pueden enviar directamente los datos al dreno. Si un nodo puede enviar sus paquetes de datos al dreno utilizando h saltos, entonces la entrega de paquetes de datos requerirá h transmisiones. Denotando con h_m el valor medio del número de saltos por paquete de datos, y sin considerar ninguna interferencia en la transmisión del radio, se tiene entonces que el máximo número de nodos que pueden ser atendidos por una WSN multi-salto con un dreno está dado por:

$$N \leq R_b \alpha_A T_R / (8 D h_m) \tag{2.1}$$

por lo tanto la capacidad de la red está limitada por el factor h_m.

En las redes de sensores inalámbricas, los nodos se deben posicionar de tal manera que puedan detectar los eventos para los cuales fueron configurados y poder transmitir la información monitorizada hacia el nodo central, en donde los datos serán procesados para actuar en respuesta al evento detectado.

La posición de los nodos sensores no necesita tener un diseño específico o predeterminado. Esto permite el despliegue de la red de manera aleatoria en áreas de difícil acceso que hayan sido afectadas por situaciones de desastre. Esto implica que los protocolos utilizados en las redes de sensores deben contar con capacidades de auto-organización. Otra característica peculiar de las WSNs es la cooperación de los

nodos sensores. Los nodos están equipados con un procesador, lo que permite que, en lugar de enviar los datos sin procesar, puedan ser pre-procesados utilizando técnicas de fusión de información. Debido a las limitaciones de los nodos sensores, el procesamiento a nivel local debe realizar cálculos simples y transmitir sólo los datos necesarios y parcialmente procesados.

Una de las más importantes limitaciones de los nodos sensores es el requisito de bajo consumo de energía. Los nodos sensores están limitados, por lo general a fuentes de energía insustituible. Por lo tanto, mientras las tradicionales redes tienen como objetivo lograr una alta calidad de servicio, las WSNs se concentran principalmente en la conservación de energía. Deberán incorporar mecanismos equilibrados que den a los usuarios finales la opción de prolongar el tiempo de vida de las redes con el coste de rendimiento más bajo o retardo de transmisión más alto.

2.3. Factores que influyen en el diseño de las WSNs

El diseño de una red de sensores inalámbrica está influenciado por muchos factores, los cuales incluyen tolerancia a fallos, escalabilidad, costes de producción, ambientes de operación, topología de la red, restricciones de hardware, medios de transmisión y consumo de energía. Estos factores son importantes porque sirven de guía para diseñar un protocolo o un algoritmo para WSNs.

2.3.1. Tolerancia a Fallos

Algunos nodos sensores pueden fallar o ser bloqueados debido a la carencia de energía, tener daños físicos o interferencia ambiental. Sin embargo, estos fallos no deberán afectar a las tareas de la WSN, debiendo mantener un mínimo de fiabilidad o tolerancia a fallos. La tolerancia a fallos es la habilidad para mantener la funcionalidad de las WSNs sin interrupciones a pesar de fallos de los nodos sensores. La fiabilidad $R_k(t)$ de los nodos de sensores es típicamente modelada en [6] usando la distribución Poisson para capturar la probabilidad de no tener un fallo dentro de un intervalo de tiempo $(0, t)$:

$$R_k(t) = exp(-\lambda_k t) \tag{2.2}$$

Donde λ_k es son la tasa de fallo del nodo de sensor k y t el periodo de tiempo.

2.3.2. Escalabilidad

El número de nodos sensores desplegados dentro del área para la monitorización de un evento puede ser del orden de cientos o miles. Dependiendo de la aplicación, el número puede alcanzar hasta un valor de millones tornándose un reto importante. En lo que respecta a la densidad, ésta puede ir desde unos pocos sensores hasta unos pocos cientos de sensores en una región, los cuales pueden estar a menos de diez metros de distancia [7]. La densidad puede ser calculada de acuerdo a [8] como:

$$\mu(R) = (N\pi R^2)/A \tag{2.3}$$

donde N es el número de nodos sensores desplegados en la región A, y R el rango de transmisión. La densidad óptima de nodos depende de la aplicación en la cual los nodos sensores son desplegados.

2.3.3. Costes de producción

Debido a que las WSNs se componen de un gran número de nodos sensores, el coste de un simple nodo es muy importante para justificar el coste total de las redes del sistema. Si el coste de la red es más elevado que el despliegue de sensores tradicionales, entonces el coste de la WSN no se justifica. Como resultado, el coste de cada nodo sensor tiene que mantenerse bajo.

2.3.4. Restricciones del Hardware

Un nodo sensor está compuesto de cuatro componentes básicos: una unidad de monitorización con sensores, una unidad de procesamiento, una unidad de transmisión-recepción y una unidad de energía (ver figura 2.1). Estos componentes también pueden tener aplicaciones que dependan de componentes adicionales, tales como sistemas de localización, un generador de energía y un generador de movimiento. Las unidades de monitorización están usualmente compuestas de dos sub-unidades: el sensor y un convertidor analógico-digital (ADC). Las de señales analógicas producidas por el sensor están basadas en la obtención del evento, y son convertidas a señales digitales por el ADC, y después procesadas dentro de la unidad de procesamiento. Uno de los componentes más importantes de un nodo sensor es la unidad de energía. La unidad de energía puede ser proporcionada por una unidad generadora de energía, tal como celdas solares.

2.4. Arquitectura de comunicación de las WSNs

Las WSNs son comúnmente desplegadas dentro de un campo de sensores como se muestra en la figura 2.3. Cada uno de estos nodos sensores tiene la capacidad de recolectar datos y encaminarlos hacia el dreno. Los datos son encaminados hacia el administrador mediante comunicación multi-salto dirigida hacia el dreno, como se muestra en la figura 2.3. El dreno se puede comunicar con el administrador vía Internet o vía satélite.

La pila de protocolos utilizada por el dreno y todos los nodos sensores se muestra en la figura 2.4. Esta pila de protocolos combina adecuadamente la energía y el encaminamiento, integra los datos con los protocolos de red, ofrece comunicación eficiente a través de medios inalámbricos, y promueve esfuerzos cooperativos de los nodos sensores. La pila de protocolos consta de la capa de aplicación, la capa de transporte, la capa de red, la capa de enlace, la capa física, el plano de gestión de energía, el plano de gestión de movilidad y el plano de gestión de tareas.

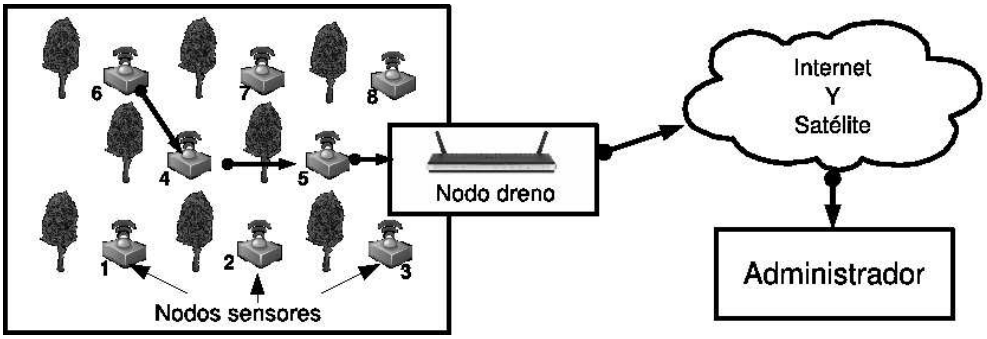


Figura 2.3: Nodos sensores desplegados en un área forestal.

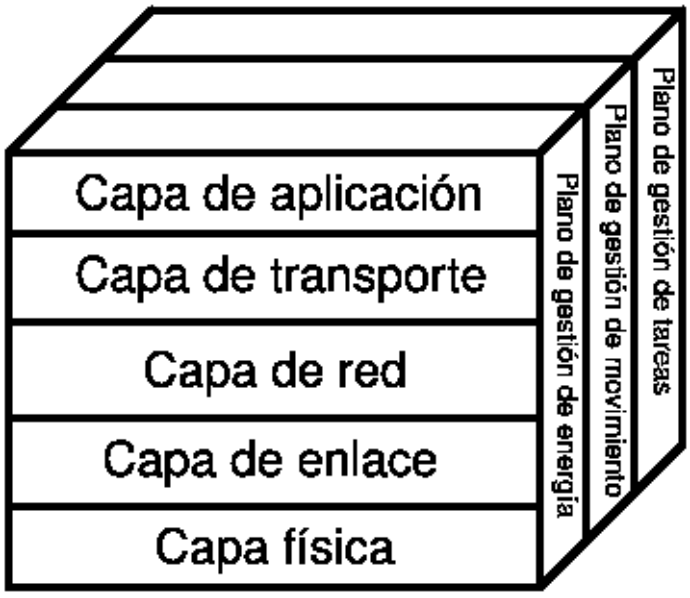


Figura 2.4: Pila de protocolos de las redes de sensores.

La capa de transporte ayuda a mantener el flujo de datos si la aplicación WSN lo requiere. La capa de red se encarga de proporcionar el encaminamiento de los datos. El protocolo MAC deberá hacer un buen uso de la energía y ser capaz de minimizar colisiones con los paquetes *broadcast* de los vecinos. La capa física atiende las necesidades de una simple pero robusta modulación, transmisión y técnicas de recepción. Además, los planos de gestión de tareas controlan la energía, el movimiento y la distribución de tareas entre los nodos sensores. Estos planos ayudan a los nodos sensores a coordinar las tareas de monitorización y, sobre todo, el bajo consumo de energía. El plano de gestión de energía administra cómo los nodos sensores usan la energía. El plano de gestión de movilidad detecta y registra el movimiento de los nodos sensores de manera que la ruta de regreso hacia el administrador siempre se conserve, y los nodos sensores puedan mantener el rastro de quienes son sus nodos sensores vecinos. El plano de gestión de tareas balancea y programa las tareas dadas en una región específica. Estos planos de gestión son necesarios, de manera que los nodos sensores pueden trabajar juntos de forma eficiente, compartiendo los recursos entre si.

2.5. Componentes de las WSNs

Una WSN se caracteriza por dispositivos de tamaño pequeño, y por la capacidad de monitorizar fenómenos ambientales a través de un conjunto sensores, así como de enviar los datos a través de transmisores/receptores. Actualmente los sensores de baja gama emplean microcontroladores de bajo coste RISC (*Reduced Instruction Set Computing*) con un pequeño programa y un tamaño de memoria de datos de cerca de 100 kb. Se puede agregar una memoria *flash* externa con largos tiempos de acceso como almacenaje secundario, con la finalidad de disminuir las restricciones del tamaño de la aplicación, impuestas por el chip de memoria interna.

Se han adoptado dos enfoques para el diseño del equipamiento de los transmisores/receptores. El enfoque más general y expansible consiste en el desarrollo de tarjetas transductoras que pueden ser agregadas a la tarjeta del microcontrolador principal a través de un bus de expansión. Otro enfoque es poner directamente el transmisor/receptor sobre la tarjeta del microcontrolador. Los transductores son adheridos o pueden ser montados si es necesario, pero las opciones disponibles son muy limitadas y generalmente afecta la escalabilidad.

Por medio de los circuitos del transmisor/receptor, una unidad de sensor se comunica con unidades cercanas. Actualmente el hardware de los sensores se basa en comunicación de Radio Frecuencia (RF). La comunicación óptica es más barata, más fácil para construir y consume menos energía que la RF, pero requiere visibilidad y direccionalidad, los cuales son extremadamente difíciles de proporcionar dentro de una WSN.

Actualmente, los nodos sensores emplean uno de dos tipos de radios. La alternativa más simple y más barata ofrece un protocolo básico de control de acceso al medio (*Medium Access Control* o MAC) denominado *Carrier Sense Multiple Aces* (CSMA), operando dentro de una banda de acceso libre (315/433/868/916 MHz) y con un ancho

de banda en el rango de 20-50 kbps. Los nuevos modelos soportan un radio 802.15.4 operando en la banda de 2.4 GHz y ofreciendo un ancho de banda de 250 kbps. El alcance del radio varía con un máximo de aproximadamente 300 metros (exterior) para el primer tipo de radio, y 10 metros para radios 802.15.4.

El tamaño de la batería usualmente determina el tamaño de los sensores, de tal manera que el tamaño del hardware existente es aproximadamente de unos centímetros cúbicos.

Un componente final es el sistema operativo, el sistema básico de software que los programadores de aplicación pueden usar para interactuar con el hardware del sensor.

2.6. El estándar IEEE 802.15.4

El estándar IEEE 802.15.4 define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (*low-rate wireless personal area network* o LR-WPAN). La ventajas de las LR-WPAN es que son de fácil instalación, transmisión de datos segura, corto alcance de operación, extremadamente bajo coste y una vida de batería razonable, mientras que mantienen una pila de protocolos simple y flexible. La actual revisión del estándar se aprobó en 2006. El grupo de trabajo IEEE 802.15.4 es el responsable de su desarrollo.

2.6.1. La capa física

La capa física proporciona dos servicios: el servicio de datos PHY y el servicio de administración PHY, interactuando con la entidad de administración de la capa física (PLME). El servicio de datos PHY permite la transmisión y la recepción de la unidad de datos de protocolo PHY (PPDU) a través del canal de radio físico.

Las características de la PHY son activar y desactivar el transmisor-receptor de radio, detector de energía (ED), indicador de calidad de enlace (LQI), selector de canal, evaluador de canal libre (CCA), y la transmisión y recepción de paquetes a través del medio físico.

La especificación de la capa física (PHY) por parte del estándar IEEE 802.15.4 define la manera en que los dispositivos pueden comunicarse entre sí sobre el canal inalámbrico, permitiendo el uso de tres bandas de frecuencia con variaciones en las tasas de transferencia de datos. Las tasas de transferencia de datos son 250kbps en la banda de 2.4 GHz, 40 kbps en la banda de 915 MHz y 20 kbps en la banda de 868 MHz. La más alta tasa de transferencia de datos en la banda de 2.4 GHz se atribuye a un esquema de modulación de orden más alto. La frecuencia más baja proporciona un rango más grande debido a menores pérdidas de propagación. La figura 2.5 muestra un resumen de las bandas de frecuencia y sus tasas de transferencia.

2.6.2. La capa MAC

La capa MAC define dos tipos de nodos: RFDs (*Reduced Function Devices*) y FFDs (*Full Function Devices*). Los FFDs están equipados con un conjunto completo

PHY (MHz)	Frecuency Band (Mhz)	Spreading parameteres		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868	868-868,6	300	BPSK	20	20	Binary
915	902-928	600	BPSK	40	40	Binary
2450	2400-2483,5	2000	O-QPSK	250	62,5	16-ary Orthogonal

Figura 2.5: Bandas de frecuencia y tasas de transferencia.

de funciones de la capa MAC, que les permite actuar como coordinador o como dispositivo final de la red. Cuando actúan como coordinador de la red, los FFDs envían balizas que permiten sincronización, comunicación y unión de los servicios de la red. Los RFDs solamente actúan como dispositivos finales de la red y están equipados con sensores actuadores, ligeros interruptores de luz, lámparas, etc. Ellos sólo pueden interactuar con un único FFD. Dos tipos principales de topologías de red son las contempladas en IEEE 802.15.4: la topología estrella y la topología punto a punto. En la topología estrella se adopta un tipo de red maestro-esclavo. Los FFDs toman un rol de coordinador PAN (red de área personal). Los otros nodos pueden ser RFDs o FFDs, y sólo pueden comunicar con el coordinador PAN. En la topología punto-a-punto, los FFDs pueden comunicar con otro FFD dentro de su rango, y puede transmitir mensajes a otros FFDs fuera del radio de cobertura, utilizando un FFD intermediario, formando una red múltiple. El coordinador PAN se utiliza para administrar la operación de la red.

Dependiendo de la configuración de red, una WPAN de baja tasa de transmisión (LR-WPAN) puede utilizar uno de los dos mecanismos de acceso al canal. Si hay un coordinador de acceso al canal, se introduce un mecanismo basado en supertrama que fragmenta el tiempo de manera que permite el acceso al canal sin colisiones. En redes sin coordinador, se utiliza el estándar CSMA-CA. Estas redes trabajan de la siguiente forma. Cualquier dispositivo que desee transmitir durante el periodo de acceso de contención, espera a que empiece la siguiente ranura de tiempo, y después determina si algún otro dispositivo se encuentra transmitiendo en esa misma ranura de tiempo. Si algún otro dispositivo se encuentra transmitiendo, el dispositivo espera un número aleatorio de ranura o indica un fallo en la conexión después de varios intentos. Una función importante de la MAC es la confirmación de recepciones exitosas de tramas de algún dispositivo. Las recepciones exitosas y las validaciones de datos o comandos MAC se confirman por medio de paquetes de reconocimiento denominados *acks*. Si el dispositivo de recepción no es capaz de recibir la información en ese momento por algún motivo, el receptor no envía ningún *ack*. El campo de control en la trama indica si se espera un *ack* o no. La trama que contiene el *ack* se envía inmediatamente después de que se hace una validación exitosa de la trama de entrada. Las tramas de sincronización (*beacon frames*) enviados por el coordinador del PAN y las tramas de

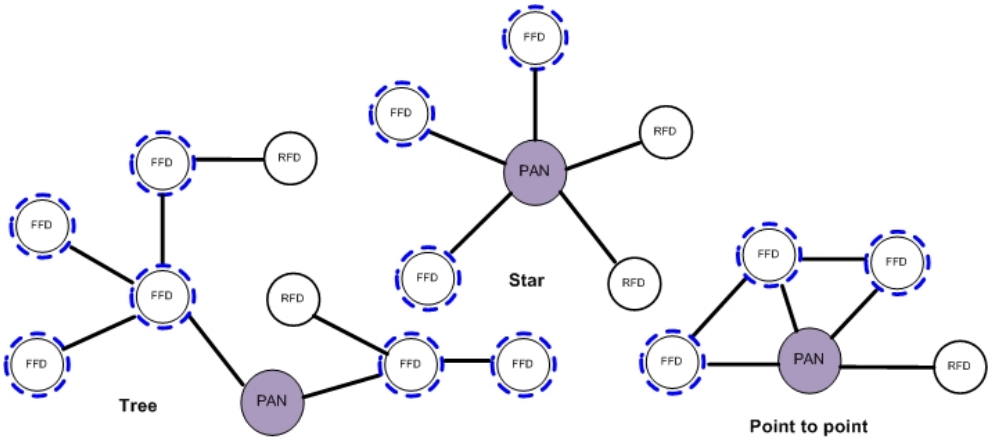


Figura 2.6: Topologías de red: árbol, estrella y punto a punto.

acks nunca son respondidas con paquetes *ack*.

Además de la transferencia de datos, la capa MAC ofrece exploración de canales y funcionalidades de asociación y disociación. El procedimiento de exploración implica el análisis de varios canales mediante el envío de solicitudes y escucha (escaneo activo realizado por FFDs) o solamente escucha (escaneo pasivo realizado por RFDs) de mensajes beacon que permiten detectar la existencia de PANs y coordinadores. La capa superior decide a que PAN se une, y más tarde pregunta a la capa MAC quien comienza el procedimiento de asociación seleccionado por la capa PAN. Esto implica enviar una solicitud al coordinador y esperar la aceptación correspondiente del mensaje. Si es aceptado por la PAN, los nodos reciben 16-bit en direcciones cortas que pueden ser usadas mas tarde en lugar de 64-bit.

2.6.3. Topologías de red

Las topologías de red soportadas por el ZigBee son tres: estrella, punto a punto y árbol.

En la topología estrella (figura 2.6 centro), la comunicación se establece entre los dispositivos y un nodo controlador central, llamado coordinador de la red de área personal (PAN). El coordinador PAN puede estar conectado a una red de alimentación, mientras que los dispositivos pueden estar alimentados por baterías. Las aplicaciones que se pueden beneficiar de esta topología son domótica, periféricos de ordenador y juguetes. Después de que un FFD se activa por primera vez, este puede establecer su propia red y llegar a ser el coordinador PAN.

En la topología punto a punto, que se muestra en la figura 2.6 (derecha), existe también un coordinador PAN. A diferencia de la topología estrella, cualquier dispositivo se puede comunicar con cualquier otro dispositivo que esté dentro de su rango de alcance. Una red punto a punto se puede auto-organizar. Entre las aplicaciones que pueden hacer uso de esta topología encontramos aplicaciones de control y monitori-

zación industrial, redes de sensores inalámbricas, monitorización de inventarios, etc. También permite múltiples saltos para encaminar los mensajes desde cualquier dispositivo a otro dispositivo dentro de la red, proporcionando de esta manera confiabilidad en el encaminamiento multi-salto.

La topología en árbol se muestra en la figura 2.6 (izquierda), y es un caso especial de las redes punto a punto, en la que la mayoría de los dispositivos son FFDs; un RFD se puede conectar a la red en árbol como un nodo hoja al final de la rama. Cualquiera de los FFDs puede actuar como coordinador y proporcionar servicios sincronizados tanto a otros dispositivos como a coordinadores. Sin embargo, sólo uno de los coordinadores puede llegar a ser coordinador PAN.

El coordinador PAN forma el primer *cluster* del árbol, estableciéndose él mismo como la cabeza del árbol o *cluster head* (CLH), con un identificador de *cluster* (CID) de cero, seleccionando un identificador PAN sin usar y enviando paquetes *broadcasting* y de *frame beacon* a los dispositivos vecinos. Un dispositivo candidato que recibe un *frame* beacon puede solicitar al CLH unirse a la red. Si el coordinador PAN permite al dispositivo unirse, este podrá agregar el nuevo dispositivo como un dispositivo hijo dentro de la lista de sus vecinos. El nuevo dispositivo que se ha unido podrá agregar el CLH como su padre dentro de la lista de vecinos y comenzar a transmitir *beacons* periódicamente, de tal manera que otros dispositivos candidatos puedan entonces unirse a la red de ese dispositivo. Una vez que la aplicación o los requerimientos de la red se cumplen, el coordinador PAN puede dar instrucciones a los dispositivos para llegar al CLH de un nuevo primer *cluster* adyacente.

2.6.4. Arquitectura del dispositivo LR-WPAN

La arquitectura del IEEE 802.15.4 está definido en capas. Cada capa es responsable de una parte del estándar y de ofrecer los servicios a las capas superiores. El dispositivo integra una PHY, la cual contiene el transmisor-receptor de radio frecuencia RF, con su mecanismo de control de bajo nivel y una sub-capa MAC que proporciona el acceso al canal físico para todos los tipos de transferencias. Las capas superiores están formadas por una capa de red, la cual proporciona la configuración de la red, el envío y el encaminamiento de mensajes y la capa de aplicación. Un control de enlace lógico en el IEEE 802.2 (LLC) puede acceder a través de la sub-capa MAC a la sub-capa de convergencia para servicios específicos (SSCS). La figura 2.7 muestra la arquitectura del dispositivo LR-WPAN.

2.7. El estándar ZigBee

La Alianza ZigBee (*ZigBee Alliance*) está formada por una asociación de industrias que trabajan en conjunto para desarrollar normas y productos. ZigBee es el nombre de la especificación de un conjunto de protocolos de comunicación inalámbrica de alto nivel, para su utilización en aplicaciones de radiodifusión digital de bajo consumo, con base en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (*Wireless Personal Area Network* o WPAN). La tecnología ZigBee está integrada en una amplia

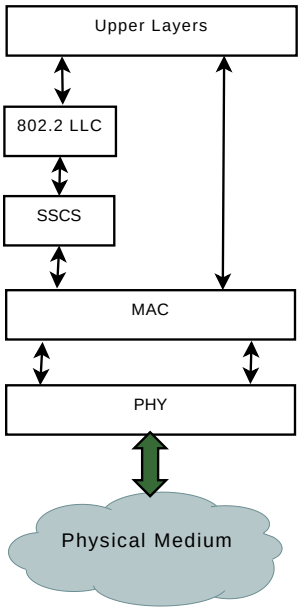


Figura 2.7: Arquitectura del dispositivo LR-WPAN.

gama de productos y aplicaciones para los consumidores de tipo comercial, industrial y gobierno. Este estándar asume el uso de las topologías estrella, árbol y punto a punto, y proporciona la estructura para la programación en la capa de aplicación. Su objetivo son las aplicaciones para redes inalámbricas que requieren comunicaciones seguras y fiables con baja tasa de envío de datos y reducido consumo energético. La gran mayoría de las WSNs usan tecnología inalámbrica basada en el estándar IEEE 802.15.4, algunas veces referenciada como ZigBee. El estándar ZigBee es una nueva tecnología desarrollada para redes de sensores inalámbricas, con las siguientes características.

- ZigBee soporta velocidades comprendidas entre 20 kb/s y 250 kb/s.
- Los rangos de alcance son de 10 m a 75 m.
- Puede usar las bandas libres ISM de 2,4 Ghz (Mundial), 868 Mhz (Europa) y 915 Mhz (EEUU).
- Pueda estar formada por hasta 255 nodos coordinadores (un coordinador por red), donde cada red podrá tener hasta 255 nodos.
- La duración la batería puede ser de hasta dos años.
- Soporta las topologías de red: estrella, punto a punto, malla y árbol
- Acceso al canal mediante CSMA/CA (acceso múltiple por detección de portadora que evita colisiones).

- Red escalable.
- Gestión automatizada de direcciones de dispositivos.

Algunas ventajas:

- Ideal para conexiones punto a punto y punto a multipunto.
- Diseñado para el direccionamiento de información y la actualización de la red.
- Opera en banda libre de ISM 2.4 GHz para conexiones inalámbricas.
- Adecuado en redes de baja tasa de transferencia de datos.
- Direccionamiento de 16 bits a 64 bits.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Proporciona mecanismos de detección de energía (ED).
- Soporta múltiples topologías de red.
- Hasta 65.000 nodos en una red.

Algunas desventajas:

- La tasa de transferencia es muy baja.
- Sólo manipula paquetes pequeños comparados con otras tecnologías.
- Tiene cobertura inalámbrica reducida porque pertenece a la familia de redes WPAN.

2.7.1. La capa de red

ZigBee identifica tres tipos de dispositivos, de acuerdo a su rol en la red. Un dispositivo ZigBee final (*ZigBee End Device* o ZED) correspondiente a IEEE RFD o FFD, actuando como un simple dispositivo que posee la funcionalidad necesaria para comunicarse con un nodo padre (el coordinador o un *router*), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es, por lo tanto, significativamente más barato. Un *router* ZigBee (*ZigBee Router* o ZR) es un FFD con la capacidad de encaminar paquetes hasta su destino final. Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código del usuario. El coordinador ZigBee (*ZigBee Coordinator* o ZC) es un FFD que gestiona toda la red, y solo hay uno en cada red. Es también el tipo de dispositivo más completo. Sus funciones son las de controlar la red y las rutas que deben seguir los dispositivos para conectarse entre ellos, por lo que requiere mayor memoria y capacidad de computación.

La capa de red ZigBee, además de la topología estrella, soporta topologías más complejas como árbol y malla. Entre las funciones proporcionadas por la capa de red están encaminamiento multi-salto, descubrimiento y mantenimiento de rutas, seguridad, capacidades para asociarse y desasociarse de una red y asignación de dirección corta de 16-bits para los dispositivos recientemente agregados, entre otras.

2.7.1.1. Descubrimiento de la ruta

El descubrimiento de ruta es un proceso necesario para establecer entradas en la tabla de encaminamiento, permitiendo así que se puedan comunicar los nodos a lo largo de la ruta. El descubrimiento de rutas en ZigBee está basado en el algoritmo AODV (*Ad hoc On Demand Distance Vector*) [9]. Cuando los nodos necesitan una ruta hacia cierto destino, emiten mensajes de solicitud de ruta (RREQ) que se propaga a través de la red hasta que llega al destino. Cada mensaje RREQ lleva un *id_RREQ* que se incrementa cada vez que se envía un nuevo mensaje RREQ. De esta manera, el RREQ ID y la dirección de origen pueden ser usadas como referencia única para descubrir la ruta. El nodo que recibe el RREQ lleva a cabo una búsqueda en la tabla de encaminamiento, para encontrar una entrada que coincida con la ruta solicitada. Si no hay ninguna coincidencia, se crea una nueva entrada en la tabla de encaminamiento. Por el contrario, si se encuentra una entrada, el nodo compara la ruta del mensaje RREQ con el valor almacenado en la entrada de la tabla de encaminamiento. Si el *num_seq* almacenado es mayor, sólo reenvía el mensaje al RREQ. De otra manera, registra el nuevo valor en la entrada de la tabla de encaminamiento.

2.7.2. La capa de aplicación

Una aplicación ZigBee consiste en un conjunto de objetos de aplicación distribuidos sobre varios nodos en la red. Un objeto de aplicación es una pieza de software que controla una unidad de hardware disponible en el dispositivo. El tipo de dispositivo ZigBee es un objeto especial, el cual ofrece servicios para los objetos de aplicación, permitiéndoles descubrir dispositivos dentro de la red y los servicios que proporcionan. Esto también proporciona comunicación, red y servicios de administración de seguridad. La sub-capa de aplicación proporciona servicios de transferencia de datos para los objetos de aplicación y los dispositivos ZigBee.

Una aplicación ZigBee deberá contar con un perfil de aplicación. Un perfil de aplicación define los formatos del mensaje y los protocolos para interactuar entre los objetos de aplicación que, en conjunto, forman una aplicación distribuida. El marco de referencia, o *framework* del perfil de aplicación, permite a los desarrolladores construir de forma independiente y vender dispositivos ZigBee que puedan interoperar con cada uno de ellos dentro de un perfil de aplicación dado. Cada objeto de aplicación encapsula un conjunto de características y proporciona funcionalidades para configurar y establecer valores de estos atributos, permitiendo notificar cuando un valor en los atributos cambie. En el contexto de un perfil, un grupo de atributos relacionados se denomina *cluster* y es identificado con un *id* numérico. Normalmente, un *cluster* representa un conjunto de interfaces de un objeto de aplicación para los otros objetos de aplicaciones.

Un perfil de aplicación especial llamado *perfil del dispositivo* se debe implementar para todos los nodos en una red ZigBee. Los perfiles de dispositivos, requieren la implementación de estos objetos y para soportar procedimientos de descubrimiento de dispositivos/servicios cuando un nodo intenta descubrir nodos existentes dentro de una red.

2.7.3. Seguridad en ZigBee

Los servicios de seguridad proporcionados por ZigBee incluyen métodos para establecimiento de claves, transporte de claves, protección de tramas y administración de dispositivos [10]. La alianza ZigBee describe las funcionalidades de seguridad basadas en un modelo de confianza abierto para un dispositivo, mediante el cual las diferentes capas de pila de comunicación y todas las aplicaciones se ejecutan sobre un dispositivo de confianza simple.

La arquitectura ZigBee incluye mecanismos de seguridad en las capas MAC, de red y sub-capa de aplicación de la pila de protocolos. Además, la sub-capa de aplicación proporciona servicios para el establecimiento y mantenimiento de conexiones seguras [10].

Los nodos sensores en una WSN están limitados en recursos de comunicación y poder computacional. Debido a esta restricción de recursos, existen mecanismos de seguridad en la red que son inapropiados para esta área. Los cifrados eficientes de datos pueden ser logrados al incrementar el coste de sobrecarga en la longitud de los mensajes. Pero, como la comunicación de radio es la función realizada por los nodos que más energía consume, la sobrecarga de comunicación se deberá minimizar para aumentar el tiempo de vida.

Los requerimientos de seguridad de las WSNs son:

- *Confidencialidad de los datos:* la confidencialidad de los datos significa mantener en secreto la información importante transmitida desde personas no autorizadas. Esta es una característica importante en los casos de las redes de sensores en las que los datos son transmitidos usando radio frecuencias, ya que cualquiera con un receptor de radio puede interceptar los datos. La confidencialidad de los datos se logra normalmente cifrando la información antes de ser transmitida, de tal manera que sólo personas autorizadas puedan descifrar dicha información. Por lo tanto, un adversario no será capaz de reconstruir la información importante, incluso si obtiene los datos transmitidos. El cifrado se clasifica en dos categorías: cifrado simétrico y cifrado asimétrico. En el cifrado simétrico, una clave secreta se comparte entre las partes autorizadas, mientras que en el cifrado asimétrico, el emisor cifra los datos con una clave pública y el receptor la descifra usando una clave privada. Un mecanismo de cifrado resistente no sólo evita reconstrucción de mensajes, sino que también evita que los adversarios decodifiquen incluso información parcial acerca de los mensajes. Esta propiedad se llama seguridad semántica, lo cual implica que el cifrado del mismo texto plano en dos tiempos distintos deberá resultar en dos textos de código diferente [11].

- *Autenticidad de los datos:* la autenticidad de los datos proporciona un medio para detectar mensajes de nodos no autorizados. Por lo tanto, evita que nodos no autorizados participen en la red. En otras palabras, la autenticación de datos permite a un receptor verificar que los datos son enviados por el emisor correcto. Esta es una característica importante en las redes de sensores, en las que un nodo adversario puede fácilmente inyectar un gran número de mensajes en la red [12] causando que otros nodos procesen estos mensajes, y así consuman sus recursos

energéticos. Por lo tanto, el receptor de estos mensajes necesita ser capaz de asegurar que los mensajes son de una fuente autorizada.

- *Integridad de los datos:* la comunicación en las redes de sensores inalámbricas están basadas en menajes de tipo *broadcast*. Los mensajes pueden ser fácilmente descubiertos o escuchados por un intruso sobre el medio inalámbrico. La integridad de los datos proporciona un medio para que el receptor del mensaje conozca si los datos han sido manipulados en el trayecto por un intruso [11]. La integridad de los datos está relacionada con la autenticación de datos desde la MAC, usada para autenticar datos y también proporcionar integridad de los mismos. El receptor de los datos calcula la MAC y la compara con una transmitida por el emisor. Si las dos capas MACs concuerdan, se asegura que los datos no fueron alterados. Si un intruso ha manipulado el mensaje, entonces la MAC calculada por el receptor no será igual a la MAC que fue inicialmente calculada por el emisor.

- *Actualización de datos:* la actualización de datos asegura que los datos recibidos son recientes, y que un atacante no manipuló el mensaje en el trayecto. Uno de los métodos más comunes para proporcionar autenticidad de los datos es usar un contador que se incrementa con cada uno de los mensajes y rechaza cualquier mensaje con valores del contador anteriores. Sin embargo, cada contenedor necesitará mantener una tabla de los últimos valores de los contadores de cada emisor. Este método puede resultar no muy factible en las redes de sensores inalámbricas donde los nodos sensores tienen memoria restringida y podrían no ser capaces de almacenarlos en una tabla, incluso para redes de tamaño moderado.

Las redes de sensores inalámbricas, como cualquier tecnología inalámbrica, son susceptibles de varios ataques de seguridad debido a los mensajes *broadcast* del medio de transmisión. Algunos de los diferentes tipos de ataques sobre las WSNs son espionaje, denegación de servicio, manipulación de mensajes, reenvío selectivo, ataques al dreno, por mencionar algunos [13] [14].

2.7.3.1. Claves de seguridad

Los dispositivos ZigBee usan *claves de enlace* y *claves de red* para una comunicación de datos segura en la red. Una clave de enlace de 128 bits, compartida entre dos dispositivos ZigBee habilitados, se usa para asegurar una comunicación *unicast* completa entre dos puntos. Por otro lado, toda la comunicación *broadcast* en la red está segura, usando una clave de red de 128 bits, la cual es compartida entre todos los dispositivos dentro de la red.

Por lo tanto, la seguridad entre dispositivos depende de la seguridad de la inicialización e instalación de estas claves. Una clave maestra se usa para generar claves de enlace. La clave maestra puede ser pre-instalada de fábrica, o incluso enviada desde un centro de confianza. Las claves de enlace y de red pueden también ser pre-instaladas de fábrica, pero estas no pueden proporcionar alta seguridad para la red. Un método

posible para obtener la clave de enlace sugerida por la especificación ZigBee, consiste en usar el protocolo de comunicación de establecimiento de claves simétricas entre dos dispositivos.

2.7.3.2. Seguridad en la capa MAC

Para proporcionar seguridad en las tramas de la capa MAC, ZigBee podría usar la seguridad de la capa MAC especificada dentro del estándar 802.15.4 [15]. La seguridad de las tramas de datos de la capa sólo proporciona seguridad para los mensajes transmitidos a un salto. Para proporcionar seguridad en mensajes que experimentan múltiples saltos, ZigBee debería confiar en la seguridad de la capa superior. La capa MAC usa el estándar de cifrado avanzado como su núcleo de algoritmos de criptografía, y describe una variedad de métodos de seguridad que usan el algoritmo de cifrado avanzado. La capa MAC procesa la seguridad de las capas superiores, las cuales configuran las claves y determinan los niveles de seguridad a usar.

Cuando la capa MAC transmite una trama con seguridad habilitada, esta busca el destino de la trama, recuperando la clave asociada con el destino, y después usa la clave para procesar la trama de acuerdo al conjunto de seguridad designado por las claves que está usando. Cada clave se asocia con un conjunto de seguridad simple y la cabecera de la trama de la capa MAC tiene un bit que identifica si la seguridad para la trama está habilitada o deshabilitada.

2.7.3.3. Seguridad de la capa de red

Similar a la capa MAC, los mecanismos de protección de las tramas de la capa de red deberán usar el estándar de cifrado avanzado. La capa de red enviará mensajes *broadcast* de solicitudes de ruta y procesará los mensajes de respuesta de ruta recibidos, para proporcionar soporte para mensajes de encaminamiento multi-salto. Los mensajes de solicitud de ruta son de tipo *broadcast,* y son enviados de manera simultánea a los dispositivos cercanos, así como los mensajes de respuesta de rutas originados por los dispositivos cercanos. Si la clave de enlace apropiada está disponible, la capa de red deberá usar la clave de enlace para asegurar las tramas de red salientes.

También puede ocurrir que la clave de enlace apropiada no esté disponible. En este caso, la capa de red deberá usar su clave de red activa para asegurar las tramas de red salientes, mientras que para las tramas de red entrantes, las claves de red activas son usadas para asegurarlas.

2.7.4. Eficiencia energética

La eficiencia energética es probablemente el problema más importante en las redes de sensores inalámbricas. Debido a que los nodos sensores trabajan durante grandes periodos de tiempo de forma independiente, es de extrema importancia desarrollar técnicas para prolongar el tiempo de vida de las baterías tanto como sea posible. El consumo de energía innecesario se debe evitar tanto en el diseño de hardware como en la programación de software a bajo nivel (sistema operativo y soporte *middleware*)

y alto nivel (aplicación). Los dispositivos de sensores comerciales proporcionan un alto nivel de flexibilidad, permitiendo a los programadores activar o desactivar varios componentes de hardware.

El transmisor/receptor de radio es el que requiere más energía de todos los dispositivos disponibles en un nodo sensor, teniendo un consumo de energía similar para las operaciones de transmisión y recepción. La razón principal del gasto de energía es la escucha pasiva, donde un nodo está escuchando el canal del radio. Otras razones son la colisión de paquetes, la escucha de paquetes destinados a otros nodos, y la sobrecarga generada por los paquetes de control [16].

2.7.4.1. Conjunto de conexión dominante

Una de las primeras conclusiones basadas en la observación de redes densas, es que muchos nodos cercanos son equivalentes desde el punto de vista de encaminamiento. El enfoque del conjunto de conexión dominante consiste en seleccionar un conjunto de nodos para construir una red principal. Los nodos estarán activos todo el tiempo, proporcionando conectividad a la red y almacenando temporalmente mensajes para los nodos vecinos que no forman parte de la red principal. Los nodos que no forman parte de la red principal duermen la mayor parte del tiempo (ahorrando energía) y periódicamente despiertan para intercambiar mensajes con sus nodos vecinos. Debido a que los nodos de la red principal consumen más energía que los otros, un protocolo de conexión dominante requiere que los nodos alternen entre los que forman parte de la red principal, y los que no forman parte de ella.

Los nodos de la red principal también pueden emplear algunos otros protocolos de eficiencia energética, para garantizar que los nodos se estén ejecutando todo el tiempo, proporcionándoles la capacidad de mantener la conectividad de la red y el intercambio de datos con los nodos vecinos que no forman parte de la red principal.

2.7.4.2. Capa MAC

Las soluciones de la capa MAC pretenden lograr ahorro de energía, excluyendo el uso de características de control de acceso al medio, y así las capas superiores en la pila de protocolos no son afectadas.

En los protocolos basados en ranuras [17] el tiempo se divide en periodos, cada uno con cierto número de ranuras de tamaño fijo. Los nodos permanecen activos dentro de cierto subconjunto predefinido de ranuras, enviando beacons para anunciar su planificación (en unidades de tiempo relativo) y escuchando las solicitudes de comunicación de los vecinos. La planificación se realiza de tal manera que cualquiera de los dos nodos vecinos eventualmente pueda escuchar a los demás.

La solución para la escucha pasiva y los problemas de contención de la capa MAC, consiste en planificar la transmisión por prioridad, de tal forma que cualquiera de los nodos conocerá exactamente cuándo deberá activar su radio sin tener colisiones. En el protocolo clásico TDMA (*Time Division Multiplex Access*) todos los nodos pueden ver a los demás nodos. Un nodo maestro comienza una supertrama, proporcionando tiempos de sincronización para el funcionamiento de la red. La supertrama contie-

ne una secuencia de ranuras que puede ser estática o dinámica. Pequeñas partes de la supertrama se usan por el nodo maestro para el control de la comunicación con los esclavos (tal como asignación de ranuras) y a los nodos esclavos para la comunicación con el nodo maestro, incluyendo solicitudes para la reserva de ranuras. Esta configuración es adecuada para redes pequeñas de un solo salto, pero su extensión a múltiples saltos posee problemas serios, incluyendo el requerimiento estricto de tiempo de sincronización.

El estándar IEEE 802.15.4 [15] soporta muchas características que, combinadas entre ellas, tienen un impacto considerable en el ahorro de energía. Sin embargo, lograr una cierta tasa de envío de datos y maximizar el tiempo de vida de los nodos individuales son objetivos de investigación frecuentes. En [18] los autores realizan un estudio sobre el esquema CSMA-CA sin beacons en redes bajo el estándar IEEE 802.15.4. Los autores observan que el esquema CSMA empleado en IEEE 802.15.4 no involucra intercambios RTS/CTS, como lo hace el IEEE 802.11. Como resultado, el CSMA-CA no ranurado (usado en modo sin beacons) es capaz de lograr una utilización del canal mayor que el CSMA ranurado (usado en el modo con beacons), permitiendo escalabilidad y auto-organización.

El modo CSMA-CA no ranurado de IEEE 802.15.4 no tiene mecanismos de ahorro de energía, y no proporciona ningún tiempo de garantía de entrega. Por otro lado, el modo ranura en el coordinador adopta un estado dormido periódicamente, logrando mayor eficiencia energética y mejor alcance con tiempos de entrega estrictos. El rendimiento de un *cluster* con modo beacons habilitado es estudiado por los autores en [19–21], tomando en cuenta el grado de nivel de servicio en términos de confiabilidad, utilización del dispositivo y rendimiento.

Una de las características más importantes de la eficiencia de energía es la posibilidad de deshabilitar el elemento transmisor-receptor, y activarlo, sólo cuando sea necesario. En modo ranura CSMA-CA ranurado, un paquete podría ser retrasado por varios periodos cuando un nodo tiene un fallo al entrar al canal, debido a la contención (especialmente en puntos de convergencia de datos cercanos como el nodo dreno), aunque todo el tráfico en la red sea relativamente bajo. Los periodos que permanecen dormidos son llamados CSD (*contention-inherited sleep deleted*).

Para aplicaciones con restricciones de tiempo, el tiempo de entrega puede ser más importante que el ahorro de energía. El modo GTS (tiempo de ranura garantizado) es un candidato potencial para lograr un rendimiento en tiempo real predecible para redes de área personal inalámbricas de baja tasa de envío de datos. Este modo ofrece la posibilidad de asignar y desasignar tiempos de ranura en una supertrama, y proporcionar garantías de servicio mínimos predecibles. Desde el punto de vista de asignación, GTS es similar a un tiempo de asignación de ranura TDMA. Una cantidad de ancho de banda reservada periódicamente se otorga para un flujo de datos dado. La cantidad de ancho de banda es determinada en relación al tiempo de ranura y a su periodicidad. El mecanismo GTS del IEEE 802.15.4 es más flexible que el TDMA, debido a que la duración del GTS puede ser ajustada dinámicamente a través de algunos parámetros. El análisis realizado por los autores en [22] proporciona una explicación completa del comportamiento del mecanismo GTS con respecto a las métricas de rendimiento y de retardo, modelando y dimensionando un cluster IEEE

802.15.4.

2.7.5. Encaminamiento

En un entorno WSN, donde los nodos pueden estar desplegados aleatoriamente, y la topología de red puede variar debido a los fallos de los sensores o las decisiones de eficiencia energética, la asignación y mantenimiento de las estructuras jerárquicas no es práctica. La sobrecarga de mensajes para mantener las tablas de encaminamiento, y el espacio de memoria necesaria para su almacenamiento, no es adecuado debido a las restricciones de energía y a los recursos en las WSNs.

Los protocolos de encaminamiento en redes de sensores inalámbricas deben cubrir las siguientes características:

- Mantener una tabla de encaminamiento razonablemente pequeña,
- Elegir la mejor ruta para un destino dado (ya sea el más rápido, confiable, de mejor capacidad o la ruta de menor coste),
- Mantener la tabla actualizada debido a fallos de nodos, cambios de posición o inserción de nuevos nodos.

La comunicación en los protocolos de encaminamiento puede ser de un solo salto o multi-salto: En el caso de un salto, es el modelo más simple y representa la comunicación directa: todos los nodos en la red transmiten a la estación base. Ante entornos de grandes dimensiones, este es un modelo caro en términos de consumo de energía, siendo típicamente inviable porque los nodos tienen un rango de transmisión limitado. Sus transmisiones no siempre pueden alcanzar la estación base ya que, tienen una distancia máxima de alcance de radio y por ello la comunicación directa no es una buena solución en muchos casos. En el modelo multi-salto, un nodo transmite a la estación base reenviando sus datos a uno de sus vecinos, el cual está más próximo a la estación base; a su vez, éste enviará a otro nodo más próximo, hasta que el mensaje llegue a la estación base. De esta manera la información viaja del nodo fuente al nodo destino saltando a través de los nodos sensores intermedios. Un gran número de protocolos utilizan este modelo, que será utilizado en los protocolos de encaminamiento propuestos en este trabajo de tesis.

2.7.5.1. Clasificación de los protocolos

Existen en la actualidad una gran cantidad de protocolos de encaminamiento aplicables a las WSNs, los cuales se pueden clasificar en tres grupos: protocolos de encaminamiento proactivos, reactivos y específicos de las WSN.

Los protocolos proactivos tratan de mantener actualizada la información de encaminamiento de cada uno de los nodos en la red. En esta categoría los protocolos requieren que cada nodo mantenga una o más tablas para almacenar la información de encaminamiento, procurando que el retardo de envió de los paquetes sea el mínimo. Las tablas de encaminamiento son intercambiadas entre los nodos vecinos cada vez que ocurre un cambio en la topología de la red. Esto hace que se consuma un mayor

ancho de banda y energía. Los protocolos que forman parte de esta categoría difieren entre ellos en base al número de tablas que utilizan y en el método por medio del cual los cambios en la estructura de la red se difunden. Ejemplos de estos protocolos de encaminamiento proactivos son: DSDV [23], CGSR [24], WRP [25] y OLSR [26].

Los protocolos reactivos crean las rutas de encaminamiento cuando los nodos tienen necesidad de enviar paquetes hasta un nodo destino. Cuando un nodo requiere una ruta hacia un destino, este inicia un proceso de descubrimiento de ruta dentro de la red. Este proceso termina cuando se encuentra una ruta o se han examinado todas las rutas posibles. Una vez que se ha establecido una ruta, se realiza un proceso de mantenimiento de ruta hasta que el destino sea inaccesible a lo largo de cada ruta o hasta que la ruta ya no se utilizada por un tiempo de expiración señalado. En contraste a los protocolos proactivos se reduce drásticamente la sobrecarga. Ejemplo de estos protocolos de encaminamiento son: AODV [27], DSR [28] y TORA [29]. Los protocolos reactivos como AODV y DSR alivian algunos de estos problemas (en realidad ZigBee utiliza un protocolo basado en AODV) pero es cuestionable para redes muy grandes ya que dependen de la inundación para el descubrimiento de rutas. También, DSR requiere la administración de *caches* de grandes rutas y encabezado de paquetes grandes para almacenar el recorrido.

Los protocolos de encaminamiento específicos para WSNs deberán ser ligeros en cuanto a consumo de energía, memoria y procesamiento, y deberán contemplar una sobrecarga mínima de mensajes. Idealmente deberán ser capaces de encaminar paquetes basados en el intercambio de información con sus vecinos y de recuperarse ante nodos que fallen y/o cambios frecuentes en la topología.

2.7.5.2. Específicos de las WSN

En este subtema se describen tres protocolos de encaminamiento específicos de las WSN. Los protocolos descritos enseguida son: el LEACH (Low Energy Adaptive Clustering Hierarchy) [30], el TEEN (Threshold-Sensitive Energy Efficient Protocols) [31] y el CTP (Collection Tree Protocol) [32].

El primer protocolo que describiremos es el LEACH (Low Energy Adaptive Clustering Hierarchy) [30] es un protocolo jerárquico conformado por clusters. La formación de estos clusters es distribuida, basada en un subconjunto predeterminado de los nodos que se eligen aleatoriamente como Clusters Head. La función de este rol consiste en comprimir la información que recibe de los nodos que conforman el cluster y enviar sólo mensaje con la información agregada a la estación base reduciendo de esta forma la cantidad de transmisiones. Se utiliza un esquema TDMA/CDMA MAC para evitar las colisiones entre los cluster e incluso intra-cluster. Este esquema no tiene que ver con la frecuencia de la adquisición de datos. Luego de un tiempo determinado se realiza la rotación del rol CH con la finalidad que sea equilibrado el gasto de energía realizando esta labor, se utiliza un algoritmo que busca que todos los nodos pasen por este rol.

El siguiente protocolo que se describe es el TEEN (Threshold-Sensitive Energy Efficient Protocols) [31] es un protocolo jerárquico conformado por clusters propuesto para aplicaciones de tiempo crítico. El proceso de adquisición de los datos es constante

en los nodos, aunque las transmisiones no son tan frecuentes. Un Cluster Head, envía a sus miembros un umbral fuerte, el cual indica el rango de valores que interesa del atributo que se mide y un umbral débil que indica la magnitud del cambio en el valor del atributo medido, que es representativo y que le indica al nodo que debe encender su transmisor y transmitir. El primero trata de disminuir el número de transmisiones permitiendo que el nodo transmita sólo cuando el atributo medido está en el rango de interés. Mientras que el débil por su parte, reduce aún más el número de transmisiones al evitar realizarlas cuando hay un cambio pequeño o no hay cambio en el valor medido. Asignar un valor pequeño para el umbral débil nos proporcionará unos valores más exactos de lo que está midiendo la red, con un costo mayor de energía. Cuando se realiza la rotación en el rol CH, son enviados los nuevos valores de los parámetros vía difusión. La principal desventaja de este esquema es que si los nodos no reciben los umbrales, no enviarán información y el usuario no recibirá datos de la red a pesar que los nodos miden su ambiente continuamente.

El tercer y último protocolo que se describe en este subtema es el CTP (Collection Tree Protocol) [32]. La recolección es el proceso inverso a la diseminación. Se trata de un proceso muy común en las WSNs con arquitecturas distribuidas. El mecanismo de recolección de TinyOS [33] proporciona un servicio de entrega de paquetes en una red multi-salto al nodo raíz de una topología en árbol. En esta topología puede haber más de un nodo raíz. En ese caso, el algoritmo se encarga de que al menos una de ellas reciba todos los datos (un nodo que envía un paquete no especifica a qué raíz está destinado) sin que existan garantías en cuanto a duplicados o desorden de mensajes. Recoger la información de una red en una estación base suele ser común en las WSNs. En general, se parte de uno o más *árboles* de recolección, cada uno de los cuales tiene como raíz un nodo que actúa como estación base. Cuando un nodo tiene datos para transmitir, los envía *árbol abajo* y continúa la recolección de los datos que recibe de otros nodos. En algunos casos el sistema debe ser capaz de inspeccionar el contenido de los paquetes (mantener una estadística, cálculos agregados, supresión de mensajes redundantes, etc).

2.7.6. Localización

El propósito de la localización es proporcionar algún tipo de ubicación de información para los nodos en una red de sensores. Esta puede ser utilizada por los algoritmos de encaminamiento y/o para identificar la ubicación de una fuente de datos según los requerimientos de la aplicación.

2.7.6.1. Coordenadas físicas

Las funciones de localización realizan la asignación física (coordenadas geográficas reales) a todos los nodos de la red de forma directa y eficaz. La mejor solución es usar un sistema de coordenadas físicas equipando a todos los nodos con un receptor GPS. Sin embargo, esta solución no se utiliza frecuentemente debido al coste de los receptores GPS, el consumo de energía, y los requisitos de tamaño. Esto también puede ser problemático si algunos nodos no reciben señal GPS (aquellos ubicados en

interiores o con obstáculos que impiden la recepción).

Una alternativa más barata para aproximarse a las coordenadas reales es utilizar algoritmos de localización que asumen que tan sólo unos pocos nodos principales tienen receptor GPS (o manualmente se dan las coordenadas correctas), y todos los demás usan protocolos de comunicación basados en radio y conectividad de datos para calcular su posición aproximada. Los algoritmos de localización pueden ser clasificados de acuerdo a su uso de técnicas de rangos para medir la distancia/posición relativa entre vecinos. Las técnicas de rango incluyen: indicador de intensidad de señales recibidas (RRSI), diferencial de tiempo de llegada (TDoA) y ángulo de llegada (AoA).

2.7.6.2. Coordenadas virtuales

Las coordenadas físicas son muy efectivas en la localización de fuentes de datos pero requieren hardware y protocolos complejos y/o costosos, y pueden introducir errores de aproximación y medidas no legibles. Además, la aproximación geográfica no necesariamente significa aproximación topológica. El encaminamiento basado en coordenadas físicas requiere mantener a los nodos fijos, y el uso de procedimientos de recuperación costosos, como los algoritmos de inundación. El objetivo de los protocolos de asignación de coordenadas virtuales, es soportar encaminamiento con un sistema de coordenadas basado en conectividad de la red.

Un algoritmo distribuido en el que los nodos calculan sus coordenadas virtuales. Rao et al [34] proponen un sistema de encaminamiento geográfico para situaciones cuando no se tiene información de la ubicación de los nodos, utilizando métodos geográficos. Como primer paso, los nodos localizados en el límite de la red aprenden que están sobre el límite en base a la distancia (en saltos) hacia un nodo en particular. Cada nodo inunda la red con mensajes *Hello*, de tal forma que todos los nodos límites descubren su distancia a todos los demás nodos, y cada uno puede a continuación inundar la red con mensajes conteniendo sus distancias.

2.7.7. Administración de datos

El objetivo final de una red de sensores es proporcionar a los usuarios datos relevantes del escenario que se está monitorizando. Por supuesto, los usuarios deberán tener una forma para indicar qué datos son relevantes utilizando un programa que interactúa con la red de sensores. El programa inyecta comandos en la red y muestra los datos devueltos por la misma.

Se pueden distinguir dos clases de aplicaciones. Una es la que involucra la detección de eventos, donde cada sensor periódicamente revisa si algunas condiciones ambientales son satisfechas localmente o concuerdan con el patrón definido. En tales aplicaciones, los nodos vecinos pueden cooperar para lograr un mejor nivel de detalle sobre las características de los eventos y de semejanza con el patrón, pudiendo los datos del evento estar almacenados en la red (para su consulta posterior) o ser directamente enviados al dreno.

La otra clase se centra en observaciones ambientales de largos periodos que continuamente desarrollan muestreo y resultados en flujos de datos. Esta cantidad extrema de datos no puede ser almacenada en la red, dados los límites de los recursos de memoria de los nodos. El flujo deberá enviarse al dreno o deberá ser descartado. La necesidad de recolectar datos desde muchos nodos distribuidos deberá ser balanceada con alto coste de comunicación. Una forma simple de reducir los mensajes es actuar en la capa de red y combinar varios mensajes hacia el dreno dentro de un mensaje grande. Esta solución sólo alivia problemas ya que los mensajes sólo pueden crecer hasta un tamaño máximo dentro de una red de sensores. La agregación de datos y el procesamiento de datos en la red es un enfoque que consiste en mover las actividades de computación desde el ordenador hacia la red [35], en lugar de sólo reenviar los datos hacia el dreno. Los nodos realizan tareas de computación y administración de datos, de tal manera que los datos solicitados por los usuarios no son extraídos desde los datos sin procesar del ordenador, sino que estos son directamente procesados por la red. Los nodos pueden hacer algún procesamiento sobre un flujo de datos (como tomar promedios temporales o funciones de computación) o combinar estos con otro flujo de datos, y finalmente producir un nuevo flujo de datos, el cual se reenvía hacia otros nodos.

2.7.7.1. Difusión directa

La difusión directa [36] es un intento reciente para definir el paradigma de la administración de datos en las redes de sensores. Un usuario solicita datos específicos, que son traducidos a un interés por cierta información enviada a una tasa específica.

La diseminación del evento se inicia con el *broadcasting* del dreno con mensajes a sus vecinos. Antes de reenviar el mensaje, cada nodo almacena la tasa de envío de datos y la importancia dentro de sus *caches,* configurando la dirección hacia la fuente del mensaje. De esta manera, la importancia se propaga a través de toda la red.

Los nodos que detectan o reciben datos relacionan una de sus *caches* de interés hacia los datos con la tasa de envío de datos asociada. La propagación de los datos de vecino a vecino finalmente llegan hasta el dreno. El dreno puede activar rutas para el envío de nuevos mensajes de interés, con una mayor tasa de envío de datos a través de las rutas seleccionadas. Los nodos sobre la rutas que no están activadas finalmente limpian su *caché* de interés hasta su tiempo de expiración. Los nodos elijen vecinos en base a la más alta calidad/tasa de recepción de datos. La activación puede ser disparada por nodos no drenos cuando ellos detectan una calidad de datos reducida desde las rutas existentes.

La ventaja principal de la difusión directa es que el intercambio de datos se basa exclusivamente en el intercambio local de intereses. No hay una ruta explícita multi-salto extremo-a-extremo, y no es necesaria para el encaminamiento y direccionamiento por toda la red. La entrega de datos multi-ruta (vía rutas múltiples reforzadas) y la reparación de rutas de datos locales (vía reforzamiento de nodo disparador) también están disponibles. Una desventaja es la carga no balanceada, ya que los nodos cercanos al dreno tienen que administrar una gran parte del tráfico de control y de datos. Otro problema está limitado por la posibilidad para procesamiento de datos dentro de la

red y agregación, ya que los diferentes datos pueden ser combinados sólo si ellos son encaminados a través de un nodo común.

2.7.7.2. El enfoque de la base de datos

El enfoque de la base de datos es un enfoque interesante y que recientemente ha ganado popularidad ya que ofrece potencialidad, aplicaciones independientes, abstracción de datos y características de manipulación para ver la red de sensores como un sistema de base de datos distribuido. El usuario formula la solicitud de datos a través de una consulta en lenguaje *SQL-like* que incluye la sintaxis para identificar frecuencias de muestreo, así como duración de la consulta [37]. La consulta de alto nivel es traducida a un conjunto de adquisición de datos (muestra, procesamiento de datos y operaciones de transferencia de datos), que deberán ser llevadas a cabo por los nodos en la red.

TinyDB [38] es una implementación de base de datos para redes de sensores desarrollada en UC Berkeley. Un lenguaje *SQL-Like* con extensiones para consultas y frecuencias de muestreo, se usa para expresar consultas sobre una tabla de sensores simples, que representa todos los datos muestreados en la red (con un registro por cada sensor que continuamente se está actualizando). TinyDB soporta operaciones de agregación espacial como se describe en [37], filtrado basado en predicados y uniones especiales tomadas sobre la relación de sensores y uno o varios puntos de almacén.

2.7.8. Confiabilidad

El problema de la confiabilidad es fundamental para las redes inalámbricas. Dado que los nodos están alimentados por baterías y la comunicación está basada en radio, los nodos pueden fallar y, temporal o permanentemente, pueden ocurrir desconexiones. Los datos recolectados por nodos individuales pocas veces son indispensables. Por el contrario, la información recolectada por varios nodos es generalmente agregada para proporcionar una mejor exactitud y relevancia.

Las estrategias de encaminamiento son inherentemente tolerantes a los fallos en los nodos y los enlaces, dada su dependencia de la información local (la cual se actualiza periódicamente). Los fallos de un nodo o enlace pueden evitar un encaminamiento correcto hacia algunos nodos, pero generalmente esto no compromete a la red completa.

2.7.8.1. Confiabilidad de transporte

La solución más simple a la confiabilidad de transporte es usar acuses de recibo para los paquetes de datos importantes. Proporcionar un servicio en la capa MAC es costoso y no es tolerante a fallos de enlace. El acuse de recibo en la capa de aplicación puede ser una solución, pero problemas como tiempos de expiración en la retransmisión, que difícilmente dependen de un número de saltos en las rutas, pueden ser difíciles de manejar.

La tolerancia a fallos y la confiabilidad fueron también tratadas en el paradigma de administración de datos de difusión directa [36], en el que múltiples rutas hacia el dreno aseguran mejor disponibilidad de datos. En [39], los autores presentan y

comparan algoritmos locales para construir rutas que no se interceptan como rutas entrelazadas. Las rutas disjuntas son evidentemente más propensas a fallos, ya que varios nodos/enlaces pueden fallar sobre una ruta sin afectar una ruta alterna. Sin embargo, las rutas disjuntas tienden a ser más grandes con respecto a la ruta óptima más corta y, consecuentemente, bastante ineficientes en cuanto a energía, excepto en escenarios de redes densas. Las rutas entrelazadas tienen mejores propiedades de eficiencia de energía, especialmente en redes dispersas, pero un fallo simple puede comprometer todas las rutas.

2.8. Aplicaciones tradicionales de las redes de sensores inalámbricas

Las redes de sensores inalámbricas pueden estar formadas por muchos tipos de sensores, tales como sísmicos, de muestreo magnético, térmicos, visuales, de infrarrojos, acústicos y de radar. Estos sensores son capaces de monitorizar una amplia variedad de condiciones ambientales que incluyen las siguientes: temperatura, humedad, movimiento vehicular, condiciones de iluminación, presión, composición del suelo, niveles de ruido, la presencia o ausencia de ciertos tipos de objetos, niveles de fuerza mecánica aplicada sobre objetos, y características tales como velocidad, dirección y tamaño del objeto. A continuación se explican diferentes aplicaciones que utilizan WSNs.

2.8.1. Aplicaciones militares

Las WSNs pueden ser una parte integral de aplicaciones militares con respecto a comandos, control, comunicaciones, computación, inteligencia, vigilancia, reconocimiento y sistemas de rastreo. El despliegue rápido, la auto-organización y la tolerancia a fallos son características de las WSNs que las hacen atractivas para el desarrollo de técnicas de monitorización para aplicaciones militares. Debido a que las redes de sensores se basan en el despliegue denso de nodos disponibles y de bajo coste, la destrucción de algunos nodos por acciones de enemigos no afectarán la operación militar tanto como la destrucción de un nodo tradicional, lo cual hace que el concepto de redes de sensores tenga un mejor enfoque para los campos de batalla. En esta área de aplicación podemos mencionar que el ejército estadounidense dispone de un sistema basado en redes de sensores, el FBCB2 (*force XXI battle command, brigade-and-below*) [40]. Teniendo como principal objetivo distinguir las fuerzas propias de los enemigos, ofrece a los soldados una visión del campo de batalla similar a la de un videojuego. En 2005 se presentó un prototipo que empleaba dispositivos MICA2 con sensores de sonido de bajo coste para la detección de francotiradores. El sistema es capaz de localizar el origen de un disparo con precisión de 1 metro y latencia de 2 segundos, con una separación de 0,4 segundos del segundo disparo [41]. Podemos encontrar otro ejemplo en la industria del armamento, actualmente en fase de desarrollo avanzado: un campo minado autoregenerable [42]. Se trata de una red *ad hoc* donde cada nodo es una mina anti-tanque. Si el enemigo abre una brecha en el campo, las minas lo perciben y tienen la capacidad de desplegarse para volver a cerrar el campo.

2.8.2. Aplicaciones ambientales

Algunas aplicaciones ambientales de redes de sensores incluyen rastreo de movimientos de pájaros, animales e insectos; monitorización de condiciones ambientales que afectan al cultivo y al ganado; irrigación; macro-instrumentos para monitorización de la tierra a gran escala, y exploración planetaria; detección biológica y/o química; precisión de la agricultura; monitorización ambiental en el contexto de marina, suelo y atmósfera; detección de incendios forestales; investigación meteorológica o geofísica; detección de inundaciones; mapeo de biocomplejidad de medio ambiente, y estudios de contaminación [8, 43–45]. Este amplio abanico de posibilidades hace que éste sea actualmente el campo de aplicación por excelencia para las redes de sensores.

2.8.3. Aplicaciones en el ámbito de la salud

Estas aplicaciones incluyen telemonitorización de datos fisiológicos en personas y administración de medicamentos dentro de un hospital, así como monitorización integral de pacientes, diagnósticos, monitorización de movimientos, y hasta monitorización y rastreo de doctores y pacientes dentro de un hospital [8, 46, 47]. Adicionalmente se han desarrollado algunas aplicaciones para monitorizar niños, la presión sanguínea y algunos signos vitales, y también para la alerta de sordos. Los prototipos usados tienen dos tipos de sensores: T-mote y SHIMMER [48].

2.8.4. Aplicaciones domóticas

Una aplicación típica de las WSNs para automatización de casa y oficina es la capacidad de encender y apagar la luz remotamente, monitorizar a los bebés que están durmiendo, o controlar la temperatura de las habitaciones. El uso de las aplicaciones WSN en casa u oficina crea un ambiente interactivo, sostenible y adaptativo para satisfacer las necesidades de las personas de todas las edades, contribuyendo a una mejor calidad de vida [5, 49]. Dentro de la monitorización y control del clima en las casas u oficinas están involucradas varias tareas, incluyendo el registro y la actuación, que son llevadas a cabo de manera rutinaria, quedando este tipo de aplicación dentro de las aplicaciones de tiempo real suave[1] [50]. Con este tipo de aplicaciones se pueden regular las variables ambientales como temperatura, humedad y luz [51].

2.8.5. Aplicaciones comerciales

Algunas de las aplicaciones comerciales existentes son monitorización de material frágil; construcción de teclados virtuales; administración de inventarios; monitorización de calidad de productos; construcción de espacios de oficina inteligentes; control ambiental en edificios de negocios; control de robots y orientación en ambientes de fabricación automática; juguetes interactivos; museos interactivos; control y automatización de procesos de fábricas; estructuras inteligentes con nodos de sensores internos;

[1] soft real-time

diagnóstico de máquinas; transporte; instrumentos de fábrica; control local de actuadores; detección y rastreo de vehículos; instrumentación de cámaras de procesamiento de semiconductores y rotación de maquinaria. Una de las aplicaciones más interesantes de esta área es un sistema de lista de compras inteligente que hace uso de redes de sensores inalámbricas. El sistema toma información de los productos utilizados en casa de acuerdo a las preferencias de las personas, e informa cuando la cantidad de ciertos productos es baja y de la caducidad de algunos de ellos [52].

2.9. Aplicaciones para monitorización de eventos críticos

Las aplicaciones WSN para la monitorización de eventos críticos se centran en la coordinación de los esfuerzos de varios nodos sensores para lograr una pronta detección de problemas dentro de un área monitorizada.

2.9.1. Trabajos que adoptan tecnología IEEE 802.15.4

En literatura podemos encontrar muchos trabajos que se centran en el comportamiento y las prestaciones de aplicaciones basadas en WSNs. Sin embargo, solo unos pocos investigadores han analizado las prestaciones del IEEE 802.15.4 en el soporte a la monitorización de eventos de tiempo crítico. Los autores en [7] realizan un estudio para evaluar varias características, tales como comunicación orientada a beacons o sin ellos (CSMA/CA), auto-configuración de la red, asociación y formación de una topología en árbol, re-localización del coordinador, y nodos huérfanos para WSNs basados en el estándar 802.15.4. Los mismos autores en [53] habían descrito algunos escenarios de aplicación para mostrar el potencial del 802.15.4, incluyendo una revisión general del estándar y evaluando la funcionalidad en el soporte de redes ubicuas.

Los autores en [1] realizaron una arquitectura para vigilancia y monitorización en minas, sin especificar la topología de red inalámbrica utilizada. Principalmente se centran en el uso de tecnología WSN evaluando el consumo de energía.

En lo referente a WSN de tiempo real, los autores en [2] evalúan el seguimiento de múltiples objetivos usando un algoritmo para fusión multisensor, que convierte la detección binaria en un reporte de posicionamiento fino usando correlación espacial. El algoritmo es aplicado al seguimiento y detección en tiempo real de un número desconocido de personas moviéndose a través de un campo exterior monitorizado por una WSN. Los autores también analizan el estándar 802.15.4 en ambientes analíticos y de simulación, para determinar en qué grado el estándar satisface los requerimientos específicos en automatización industrial de tiempo real. Los autores en [54] describen y evalúan una arquitectura para comunicación en tiempo real en redes de sensores inalámbricas de gran escala, para propósitos de monitorización y control. Los autores en [55] presentan la arquitectura de VigilNet, un sistema de redes de sensores a gran escala, los cuales rastrean, detectan y clasifican objetivos de manera oportuna y mediante un uso de energía eficiente. VigilNet se utiliza en operaciones militares, donde

los eventos de interés suceden con una tasa relativamente baja y dónde la duración de los eventos significativos es muy corta.

2.9.2. Aplicaciones con requisitos de tiempo real

En las aplicaciones de las WSNs que forman parte de esta área, es muy importante el tiempo de respuesta. Por ejemplo, en los eventos donde un sensor o conjunto de sensores detecten el cambio en algunas de las variables que están detectando, deberán enviar la información recolectada hasta el dreno en el menor tiempo posible, de tal manera que las acciones de respuesta al evento detectado deban ser distribuidas de la forma más rápida y fiable posible. Este modelo se debe usar principalmente en aplicaciones que requieran respuesta en tiempo real, por ser eventos críticos. En [56] los autores proponen un protocolo para servicios de tiempo real (CRTS), los cuales ofrecen diferentes herramientas para recolección de eventos simultáneos en tiempo real, permitiendo para este propósito diferentes niveles de QoS de manera dinámica. En [57] desarrollan e implementan una WSN para monitorizar el movimiento y posición de personas en interiores. Los datos del movimiento son procesados en nodos móviles y transferidos a una red estática.

2.9.3. Monitorización de la propagación de gas y fuego

Para la detección de la propagación de gas y fuego, los nodos sensores pueden ser desplegados densamente, aleatoriamente y estratégicamente en un bosque. Los nodos sensores pueden revelar el origen exacto del fuego o la presencia de gas a los usuarios finales, antes de que el fuego o gas se propague de manera descontrolada. Miles de nodos sensores pueden ser desplegados. También ellos pueden ser equipados con métodos eficientes de uso de energía, tales como celdas solares, debido a que los sensores pueden ser desatendidos por meses o incluso años. Los nodos sensores pueden colaborar entre ellos para una monitorización distribuida y superar obstáculos, tales como árboles y piedras.

En [58] los autores describen una red de sensores para monitorización de gas en tiempo real, proporcionando un método de auto-calibración para garantizar que las redes de sensores no requieran mantenimiento, y así recolectar datos a través de los sensores que miden la concentración de contaminación. En particular, los autores describen el sistema de sensores para medir bióxido de nitrógeno, un agente altamente oxidante. En [59] los autores hacen uso de redes bayesianas como forma de aprendizaje no supervisado y detección de eventos anormales en redes de sensores; el objetivo de la aplicación es la monitorización de gas en minas de carbón.

En el caso de los incendios forestales, una red de sensores inalámbrica se puede desplegar para detectar fuegos durante sus primeras etapas, y así actuar tan rápido como sea posible. Para el despliegue de los nodos sensores se deberán tratar de cubrir todos los puntos posibles de inicio del fuego, de tal manera que los nodos sensores puedan detectar el evento de fuego antes de que se propague. Similarmente, cuando un evento relacionado con el fuego, tal como el incremento de temperatura, un centro de control deberá ser notificado con los datos recogidos, así como con la ubicación

del fuego. En general, los sistemas de detección de fuego basados en WSNs deberán operar de manera cercana al tiempo real para evitar al máximo los daños causados por este tipo de siniestros.

Otra de las investigaciones relacionadas con la detección de fuego forestal se presenta en [60], donde se propone una WSN para detectar fuegos en tiempo real usando redes neuronales para el procesamiento de datos dentro de la red. En este artículo se propone el uso de un gran número de nodos sensores, que son desplegados en un bosque. Los nodos recolectan los datos medidos tales como temperatura y humedad, y los envían a un nodo *cluster* para procesar los datos colaborativamente de cara a construir una red neuronal. Otra propuesta para monitorizar fuego forestal fue llevada a cabo en [61]. En este trabajo se incluyen técnicas en tiempo real y el hardware utilizado está basado en el chip CC2430 recientemente lanzado por la compañía Chipcon. Los parámetros medidos en tiempo real son la temperatura y la humedad en algunas regiones de prueba. En el artículo se describe la arquitectura implementada; sin embargo, no se muestran los resultados obtenidos con el sistema propuesto. Dentro de las propuestas para monitorizar fuegos forestales también se encuentra el trabajo propuesto en [62], en el cual se diseña e implementa un sistema de vigilancia basado en redes de sensores inalámbricas para las montañas del sur de Corea. El sistema propuesto mide la temperatura, la humedad y detecta el humo. Una aplicación web analiza los datos recolectados por los nodos sensores, permitiendo al sistema el envío de alarmas en tiempo real cuando ocurre un fuego forestal.

2.9.4. Seguimiento de intrusos

El objetivo de las aplicaciones de seguimiento de intrusos es monitorizar el movimiento de un objetivo dentro de un área. Li *et al.* en [63] presentan el tema de detección y seguimiento de un objetivo simple en una WSN, usando la coordinación entre protocolos de encaminamiento y algoritmos de localización. Los autores extienden este proyecto al seguimiento de múltiples objetivos. Miyashita en [64] realiza el estudio de las características del despliegue de una WSN para la detección de objetivos, identificando los mejores tipos de sensores para ser usados en este tipo de aplicaciones en ambientes de redes densas, cuando el enfoque es sobre objetivos del tipo humano o vehículos. Cao *et al.* en [65] establecen la relación entre los parámetros y atributos de un sistema de vigilancia aplicado a objetivos en movimiento y estáticos. Los autores adoptan el modelo de planificación *duty-cycle* de forma individual, nodos no sincronizados, permitiendo a los nodos dormir y despertar periódicamente. Chen *et al.* en [2] desarrollaron una aplicación para control y vigilancia en WSNs de tiempo real a gran escala, usando algoritmos de seguimiento de múltiples objetivos los cuales fueron el resultado de combinar el algoritmo de fusión multisensor y el de MCMCDA (*Markov Chain Monte Carlo Data Association*). He *et al.* en [66] proponen un sistema para monitorizar la eficiencia de la energía, para uso en aplicaciones militares como un sistema de vigilancia que es capaz de operar por largos periodos de tiempo. En este sistema los autores evalúan las prestaciones de una red equipada con 70 MICA2 motes con magnetómetros de dos ejes. Los mismos autores desarrollaron VigilNet [55], un sistema WSN de tiempo real a gran escala que permite detectar,

rastrear y clasificar objetivos en un tiempo razonable, y haciendo un uso eficiente de la energía. VigilNet es un sistema diseñado para operaciones militares espontáneas en áreas remotas, donde los eventos de interés suceden con poca frecuencia y son de corta duración, tales como los eventos relacionados con intrusos.

2.10. Sumario

Las WSNs pueden estar integradas por un gran número de nodos sensores, los cuales son densamente desplegados en la zona dónde ocurrirán los posibles eventos a monitorizar, dependiendo de la aplicación para la que hayan sido programados. La posición de los nodos sensores no necesita ser diseñada o determinada *a priori*. Esto permite el despliegue en terrenos inaccesibles en situaciones de desastre. Una característica importante de las WSNs es que los protocolos y algoritmos deben poseer la capacidad de auto-configuración. Otra característica peculiar de las redes de sensores es la cooperación de los nodos sensores. Algunos nodos sensores, en lugar de enviar los datos sin procesar, se encargan de fusionarlos. De esta manera, los nodos sensores tienen capacidad de procesamiento a nivel local para realizar cálculos simples, y transmitir sólo los datos necesarios y parcialmente procesados.

El diseño de una red de sensores inalámbrica está influenciada por muchos factores, los cuales incluyen: tolerancia a fallos, escalabilidad, costes de producción, ambientes de operación, topología de la red de sensores, restricciones de hardware, medios de transmisión y consumo de energía. Estos factores son importantes porque ellos sirven como una guía para diseñar protocolos o algoritmos para WSNs. Además, la influencia de estos factores ha de tenerse en cuenta a la hora de comparar diferentes esquemas.

Los protocolos de encaminamiento para WSNs deberán ser ligeros en cuanto a consumo de energía, requisitos de memoria y procesamiento, debiendo además introducir una mínima sobrecarga de paquetes de control. Idealmente deberán ser capaces de encaminar paquetes basados en el intercambio de información con sus vecinos, y deberán tener capacidad de recuperar los nodos que fallen, realizando los cambios necesarios en la topología para un funcionamiento eficiente.

Capítulo 3

Modelado y seguimiento de eventos críticos

En este capítulo se describen los modelos de propagación de gas y fuego, así como los patrones de movilidad de posibles intrusos que acceden a un área monitorizada. Además, se describe una herramienta para WSNs que permite modelar fácilmente el comportamiento de este tipo de eventos. También se incluye aquí la explicación de los algoritmos de agregación de datos implementados, así como los algoritmos que realizan la reconstrucción de eventos.

3.1. Introducción

La disponibilidad de información precisa y oportuna sobre las condiciones ambientales en el caso de algunos eventos críticos, tales como la detección de gas o fuego, es de gran importancia por los posibles desastres que se pueden prevenir en estos casos. Tomando en cuenta las características de este tipo de eventos, se ha desarrollado una herramienta modeladora de eventos que integra desde la interfaz gráfica de usuario, un visualizador de eventos, el motor generador de eventos y el motor generador de tráfico.

La herramienta propuesta permite realizar diferentes tipos de experimentos simplemente variando los parámetros de entrada de los diferentes modelos.

El análisis de los datos de salida del simulador es también fundamental. Así, a partir de esos datos se analizan las prestaciones de la red, además de la precisión de la reconstrucción de los eventos, mediante algoritmos de agregación y procesado de los datos recolectados.

3.2. Modelado de la propagación de gas y fuego

En literatura especializada podemos encontrar varios artículos que se centran en modelos de propagación de fuego [67] y gas [58, 68]. Sin embargo, la mayoría de los modelos propuestos se centran en las características propias de cada escenario específico, sin centrarse en un modelo más genérico y útil para usar en otros campos de investigación relacionados.

En esta sección se proponen modelos de propagación para gas/fuego, que son bastante representativos para el estudio de eventos en tiempo-real en WSNs. El usuario puede variar la velocidad del proceso de propagación de acuerdo a diferentes valores de entrada al proceso, permitiendo así modelar diferentes condiciones.

3.2.1. Propagación en interiores

La propagación de gas o humo relacionado con el fuego en escenarios interiores es un proceso relativamente sencillo de modelar, dependiendo normalmente del volumen de aire disponible y, en menor medida, de los obstáculos existentes. Para el tipo de eventos utilizados en esta tesis, se consideró adecuado usar un modelo para propagación interior de gas/fuego basado en un patrón de expansión circular, en el que el usuario selecciona la velocidad de propagación (s) con la que el radio del círculo (R) se incrementa. Así, para la propagación de gas/humo, se utiliza la siguiente ecuación:

$$R_g(t) = s \cdot t \tag{3.1}$$

Este modelo es adecuado para describir la propagación de gas y humo dentro de escenarios interiores sin obstáculos. El valor de s deberá estar relacionado con la tasa de generación de gas/humo y las condiciones volumétricas del escenario. En el caso de fuego, el objetivo es modelar el proceso de propagación de llamas y humo. Con este propósito, adicionalmente se introdujo el coeficiente ψ, utilizado para relacionar la velocidad de propagación y del humo. Este valor puede ser ajustado de acuerdo al criterio de los usuarios, y dependerá del escenario específico y del tipo de material que está siendo modelado. El modelo de propagación de llamas está definido de acuerdo a la siguiente ecuación:

$$R_f(t) = \psi \cdot s \cdot t, \psi \epsilon [0, 1] \tag{3.2}$$

Por tanto, cuando se trate de caracterización de propagación de fuego interior, es posible usar simultáneamente las ecuaciones 3.1 y 3.2.

3.2.2. Propagación en exteriores

La propagación en exteriores de fuego y gas difiere significativamente de los correspondientes procesos en espacios interiores. En particular, el patrón de propagación de fuego y gas dentro de escenarios exteriores depende enormemente de la velocidad y dirección del viento. En el caso de fuego, el proceso de propagación de la llama

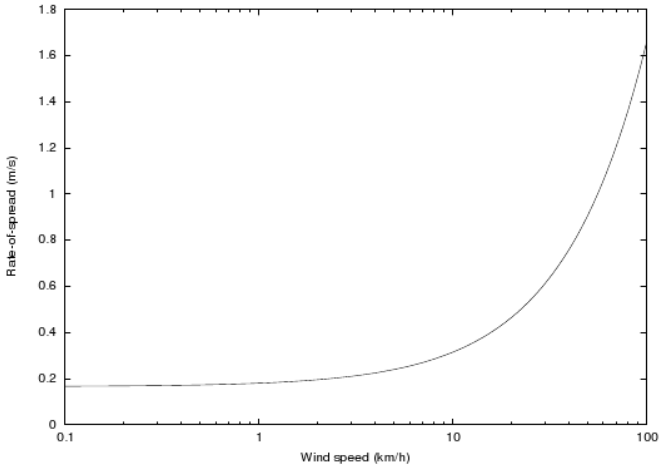


Figura 3.1: Tasa de propagación (ROS) para fuego de pasto con diferente velocidad del viento.

depende bastante de los materiales que están siendo incinerados y de su grado de proximidad. Además, debemos remarcar que, al contrario de la propagación en interiores, la propagación exterior de gas/humo no está limitada volumétricamente, por lo tanto la tasa de propagación es más baja que el modelo de propagación en interiores. También, en el caso de fuego, los procesos de propagación de las llamas y el humo están fuertemente relacionados. Específicamente, este trabajo se centra principalmente en el estudio de incendios forestales, el cual es una clase de fuego en la que se considera la existencia de pasto, arbustos y árboles. Una característica importante de espacios exteriores es que la escala de tiempo de propagación del fuego y la escala de tiempo de árboles quemados se pueden considerar independientes.

A partir del trabajo de investigación y los modelos presentados en [69], se propone modelar la propagación externa de fuego/gas utilizando una expansión elíptica, donde uno de los focos se fija en el origen del evento de fuego/gas, y el otro foco se mueve de acuerdo con la velocidad del gas o fuego frontal, normalmente conocido como ROS (*Rate of Spread*) o tasa de propagación.

Partiendo de una ecuación general de elipse paramétrica en (h, k):

$$\begin{cases} x = h + a \cdot cos\, t \\ y = k + b \cdot sin\, t \end{cases} \tag{3.3}$$

aplicamos rotación a la matriz

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\,\theta & -sin\,\theta \\ sin\,\theta & cos\,\theta \end{bmatrix} \tag{3.4}$$

obteniendo así una ecuación que permite rotar la elipse un determinado ángulo θ:

$$\begin{cases} x = h + a \cdot cos\ t \cdot cos\ \theta - b \cdot sin\ t \cdot sin\ \theta \\ y = k + b \cdot sin\ t \cdot cos\ \theta + a \cdot cos\ t \cdot sin\ \theta \end{cases} \tag{3.5}$$

Con respecto a las características de la elipse, las propiedades que se han considerado son la posición del centro, la tasa de crecimiento y la excentricidad.

Comenzando por la primera propiedad (posición del centro) definida por las coordenadas (h, k). Si queremos modelar un fuego inicial en uno de los focos de la elipse en la posición (x_c, y_c), entonces se debe reemplazar (h, k) por $(x_c + c \cdot cos\ \theta, y_c + c \cdot sin\ t)$. Esto significa que el centro de la elipse se mueve linealmente en el tiempo, como se esperaba.

Con respecto a la tasa de crecimiento, esta dependerá mucho de la velocidad del viento. En [67] los autores derivan una expresión que relaciona a la velocidad del viento (V) con la velocidad del fuego frontal (r_w) para incendios forestales, y que es la siguiente:

$$r_w = r_0(1 + c_f \cdot V) \tag{3.6}$$

En esta expresión, r_0 representa la tasa de propagación (*Rate-of-Spread*, ROS) sin viento y toma el valor de 0,165m/s, mientras c_f es un coeficiente que permite relacionar la velocidad del viento con el ROS, y toma el valor de 0,324, ambos de acuerdo a [69]. La figura 3.1 muestra, para velocidades de viento hasta 100 Km/h, los valores correspondientes al ROS.

En el modelo propuesto el fuego se inicia en uno de los focos de la elipse. Una vez que el fuego se propaga desde este punto y a lo largo del eje principal de la elipse, podemos derivar la siguiente expresión:

$$a + c = r_w \cdot t \tag{3.7}$$

donde a representa la longitud del semieje mayor, y c la distancia desde el foco al centro de la elipse.

Para terminar la caracterización de la elipse, debemos definir el valor de su excentricidad. Para obtenerlo, se consideró que, en un incendio forestal, la velocidad de propagación transversal se mantiene con respecto a un escenario sin viento. Por tanto, definimos:

$$b = r_0 \cdot t \tag{3.8}$$

donde b se refiere a la longitud del semieje menor de la elipse. Combinando las expresiones 3.7 y 3.8 obtenemos el valor de la excentricidad para la elipse:

$$e = \frac{r_w^2 + r_0^2}{r_w^2 - r_0^2} \tag{3.9}$$

la cual nos permite definir completamente la elipse que representa la propagación del fuego en cualquier momento en el tiempo.

Con respecto a la propagación del humo en *incendios forestales*, estará fuertemente asociado al área incinerada. En particular, consideramos que cubrirá un área que es λ por ciento mayor que la zona incinerada.

Para una excentricidad fija e, el área de la elipse es proporcional al valor del semieje mayor (a). Así, para el humo relacionado con el fuego, consideramos que la detección del humo se puede hacer dentro de un perímetro también definido por una elipse con un semieje mayor de: $(1 + \lambda) \cdot a$, y la misma excentricidad (e) que la elipse que contiene la zona incinerada.

3.2.3. Generación de eventos basados en gas/fuego

En las secciones 3.2.1 y 3.2.2 se presenton los modelos teórico que permiten describir la propagación de eventos de gas/fuego en escenarios interiores y exteriores. Estos modelos fueron integrados en la herramienta de generación de eventos en tiempo-real, y así como en la generación de eventos basados en intrusos. La información que ellos proporcionan se combina con la posición del sensor tomada de la traza ns-2 para generar eventos en tiempo-real. Estos eventos consisten en las definiciones de inicio/parada del tráfico de cada uno de los nodos, y usando un formato compatible con ns-2.

Los sensores se activan cuando se encuentran físicamente dentro del área del modelo de gas/humo. Por ejemplo, si un sensor se localiza en el punto S_i y los dos focos de la elipse usados para representar el área afectada en el tiempo t están localizados en las coordenadas F_1^t y F_2^t, el sensor se activa si se satisface la siguiente condición:

$$\|F_1^t - S_i\| + \|F_2^t - S_i\| \leq 2 \cdot a \cdot (1 + \lambda) \tag{3.10}$$

En el caso de propagación de fuego, los sensores dentro del área incinerada (las cuales entran en contacto con el fuego) son considerados destruidos, y dejan de formar parte de la WSN. Regresando al modelo de incendios forestales propuesto, todos los sensores que satisfacen la condición:

$$\|F_1^t - P\| + \|F_2^t - P\| \leq 2 \cdot a \tag{3.11}$$

son incinerados, y no participan más en términos de generación y encaminamiento de tráfico. Por lo tanto, los sensores inicialmente generan tráfico para advertir sobre la presencia de humo y, después de un cierto tiempo, son apagados para modelar su destrucción.

Presentamos como ejemplo gráfico la salida de eventos de propagación de gas interior y fuego exterior. La figura 3.2 muestra la evolución de un proceso de propagación de gas en diferentes instantes de tiempo considerando una velocidad de propagación de 1 m/s. Todos los sensores (representados como puntos) dentro del límite del círculo correspondiente se consideran activados, generando tráfico de advertencia.

La figura 3.3 muestra una imagen instantánea del evento de propagación de fuego forestal considerando que el fuego se originó en el punto etiquetado como *Focus*, y después se propaga en un ambiente externo donde la velocidad del viento es de 40 km/h y el ángulo para el viento es de 30^0. En el instante de tiempo representado ($t = 400s$) todos los sensores dentro del interior de la elipse se consideran destruidos,

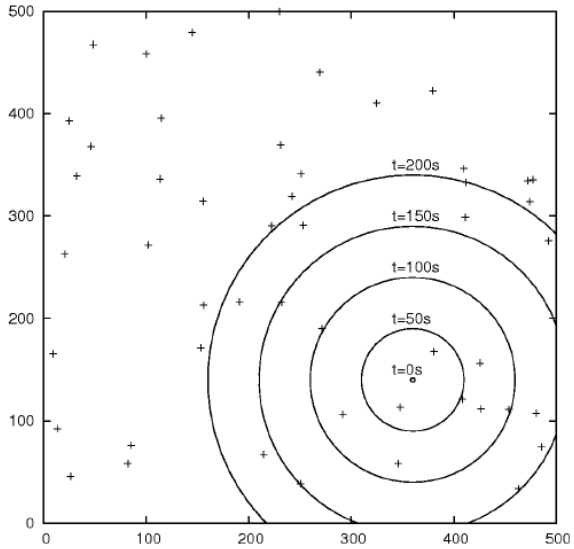


Figura 3.2: Evolución del proceso de propagación de gas en un escenario interior ($s = 1m/s$).

mientras que los sensores localizados en el área intermedia entre los límites de ambas elipses están activos, generando tráfico de advertencia hacia un dreno.

3.3. Modelado de los patrones de movilidad de intrusos

En la bibliografía relacionada con las redes *ad hoc* móviles podemos encontrar varias propuestas para describir la movilidad de los usuarios [70]. Sin embargo, la mayoría de ellas son inadecuadas para nuestros propósitos debido a que no se centran en el patrón de movilidad de una persona (el intruso), ya que en este caso la persona está normalmente moviéndose rápidamente hacia una posición específica (un punto de interés), posiblemente estando consciente de que estará siendo rastreado. Por esa razón, procedemos a modelar el movimiento del intruso de acuerdo a tres diferentes patrones de movilidad. En aras de generalizar, y para simplificar la inclusión de cualquier modelo de movilidad en nuestra herramienta, también incluimos un cuarto modelo de movimiento genérico, el cual solo restringe la ruta de entrada que está teniendo el intruso, a que sea definida en términos de una función paramétrica en 2-D o 3-D.

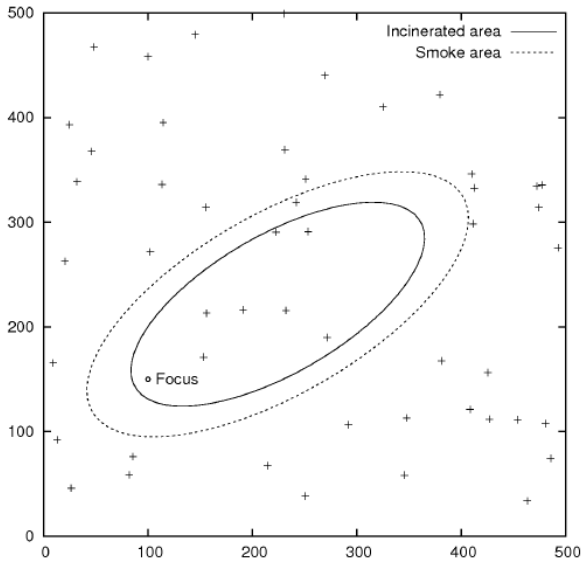


Figura 3.3: Instantánea de un evento de fuego forestal en el tiempo $t = 400s(V = 40km/hy\theta = 30^0)$.

3.3.1. Modelo de movimiento recto

El modelo de movimiento recto es el modelo mas simple, el cual representan el movimiento de un intruso dentro de un área monitorizada.

En un tiempo t_0 el intruso entra al área protegida por la WSN en una cierta posición dentro del límite del escenario; se mueve hacia un punto de interés (x_i, y_i) a una velocidad s y con un ángulo de ataque θ en el rango $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, y continúa con un movimiento recto hasta que salga del área vigilada.

Dado que el generador de eventos de WSN propuesto necesita la descripción de movimiento en función del tiempo, la descripción matemática de todos los modelos deberá ser paramétrica, usando el parámetro t para representar el tiempo (en segundos) En el caso de movimiento recto, la expresión de referencia es:

$$\begin{cases} x(t) = s \cdot cos(\theta \cdot t) \\ y(t) = s \cdot sin(\theta \cdot t) \end{cases} \tag{3.12}$$

En este caso, el modelo analítico deberá ser cambiado para cruzar el punto de interés (x_i, y_i) y fijar el tiempo de la entrada del intruso al área protegida a t_0. Se requiere determinar simplemente el tiempo $(t^{'})$ en que el intruso cruza del eje x o el eje y, y a continuación se deriva la expresión apropiada. En este caso, el resultado de la ecuación usada en nuestro modelo ha sido:

Algoritmo 3.1 *Random way-point* descripción de movilidad base.

1 Generar coordenas iniciales (x_0, y_0) aleatoriamente
2 Determinar coordenadas (x_i, y_i) de un nuevo punto de interés
3 Hacer $t'_{prev} = t'$
4 Calcular el tiempo de llegada del nuevo punto de interés: $t' = t'_{prev} + \frac{\sqrt{(y_i - y_0)^2 + (x_i - x_0)^2}}{s}$
5 Calcular el ángulo de movimiento: $\theta = arctan\frac{y_i - y_0}{x_i - x_0}$
6 Moverse al tiempo t' de acuerdo a ecuación: $(x(t), y(t)) = (x_0 + s \cdot cos[\theta \cdot (t - t')], y_0 + s \cdot sin[\theta \cdot (t - t')])$
7 Hacer $(x_0, y_0) = (x_i, y_i)$
8 Ir a paso 2

$$\begin{cases} x(t) = x_i + s \cdot cos[\theta \cdot (t - t_0 - t')] \\ y(t) = y_i + s \cdot sin[\theta \cdot (t - t_0 - t')] \end{cases} \tag{3.13}$$

donde el valor de t' depende de la posición del punto de interés, la velocidad del intruso y el ángulo de ataque actual:

$$t' = max\left(\frac{x_i}{s \cdot cos\theta}, \frac{y_i}{s \cdot sin\theta}, \frac{y_{max} - y_i}{s \cdot sin\theta}\right) \tag{3.14}$$

Observe que el último termino $\frac{y_{max} - y_i}{s \cdot sin\theta}$ se aplica cuando el intruso entra al escenario desde la parte superior (coordenada $y = y_{max}$).

3.3.2. Modelo *Random way-point*

El modelo *random way-point* es una generalización del modelo de movimiento recto, en el que los intrusos se están moviendo continuamente dentro de un área monitorizada. Este modelo ha sido ampliamente utilizado y adoptado para modelación de usuarios en redes inalámbricas *ad hoc* [71]. Nosotros nos centramos en un solo intruso que se mueve continuamente hacia diferentes puntos de interés. Por lo tanto, en el tiempo t_0, el intruso entra al área protegida por la WSN en cualquier posición dentro del escenario. A continuación se mueve hacia el primer punto de interés (x_{i_0}, y_{i_0}) en movimiento recto a una velocidad s y, a la llegada, selecciona otro punto de interés (x_{i_1}, y_{i_1}), avanzando hacia adelante nuevamente en movimiento recto y a la misma velocidad. El proceso continúa de esta manera hasta que se alcanza el tiempo de parada.

El algoritmo 3.1 ofrece una descripción formal del patrón de movilidad del intruso de acuerdo a este segundo modelo. Note que, cada vez que el intruso llega a su destino actual, se calcula un nuevo destino y el proceso se inicia de nuevo.

3.3.3. Modelo de movimiento genérico

Hasta ahora los modelos desarrollados han sido caracterizados por eventos de movimiento recto. Esa característica hace que sea fácil de aplicar una velocidad fija para el intruso como se desee.

Algoritmo 3.2 Modelo de movilidad genérico para la descripción de los parámetros de movimiento.

1 Seleccionar una función paramétrica $(x(\tau), y(\tau))$ para describir el patrón de movimiento del intruso.
2 Determinar la ecuación respectiva para la velocidad $v(\tau) = \sqrt{x^{'}(\tau)_2 + y^{'}(\tau)_2}$
3 Hacer $\tau_{ac} = 0, t^{'} = 0$ y seleccionar la coordenada inicial (x_0, y_0).
4 Hacer $\tau_{ac} = \tau_{ac} + \frac{\Delta d}{v(\tau_{ac})}$
5 Calcular el nuevo destino: $(x_i, y_i) = (x(\tau_{ac}), y(\tau_{ac}))$
6 Hacer $t^{'}_{prev} = t^{'}$
7 Calcular el tiempo de llegada al nuevo punto de interés: $t^{'} = t^{'}_{prev} + \frac{\sqrt{(y_i - y_0)^2 + (x_i - x_0)^2}}{s}$
8 Calcula el ángulo de movimiento: $\theta = arctan\frac{y_i - y_0}{x_i - x_0}$
9 Moverse al tiempo $t^{'}$ de acuerdo a: $(x(t), y(t)) = (x_0 + s \cdot cos[\theta \cdot (t - t^{'}_{prev})], y_0 + s \cdot sin[\theta \cdot (t - t^{'}_{prev})])$
10 Hacer $(x_0, y_0) = (x_i, y_i)$
11 Saltar al paso 4

Ahora buscamos desarrollar un método que permita la integración de nuestra herramienta en cualquier modelo de movilidad. Tomamos como ejemplo un escenario donde el intruso sigue una curva cuando se está moviendo dentro del área monitorizada. Consideramos que en un tiempo t_0 el intruso entra al área protegida por la WSN en la posición (x_0, y_0) dentro del escenario; se moverá hacia un punto de interés (x_i, y_i) a una velocidad s de acuerdo a una ecuación polinomial de segundo orden, y continuará el movimiento hasta que salga del área vigilada. La representación paramétrica de este patrón de movimiento es el siguiente:

$$\Big(x(t), y(t)\Big) = (t, a \cdot t^2 + b \cdot t + c) \tag{3.15}$$

En este caso, la obtención de la velocidad en función de tiempo puede lograrse mediante el cálculo de la derivada de $x(t)$ y $y(t)$:

$$v(t) = \sqrt{x^{'}(t)^2 + y^{'}(t)^2} = \sqrt{1 + (2a \cdot t + b)^2} \tag{3.16}$$

Como se esperaba, los valores de $v(t)$ variarán enormemente con el tiempo, lo que complica el proceso de fijación de la velocidad en un determinado punto, usando solo procedimientos matemáticos. Este problema se acentúa cuando se trata con funciones matemáticas más complejas, lo cual impide su utilización en la herramienta diseñada.

Para hacer frente a este problema hemos propuesto una novedosa solución basada en un enfoque doble de dominio del tiempo, combinada una técnica diferencial multi-segmento. La técnica está orientada a la representación de funciones. Requerimos un dominio de tiempo doble para separar el tiempo actual t de la variable de parametrización, que vamos a representar como τ para evitar confusión.

El algoritmo 3.2 describe los diferentes pasos involucrados en este proceso. A partir de la función paramétrica definida por el usuario y la función de velocidad derivada $(v(\tau))$, de forma iterativa avanzamos por un instante de tiempo obtenido a partir de una distancia predefinida Δd, cuyo valor puede ser ajustado para poder obtener el grado de exactitud deseado dentro del proceso de representación multi-segmento

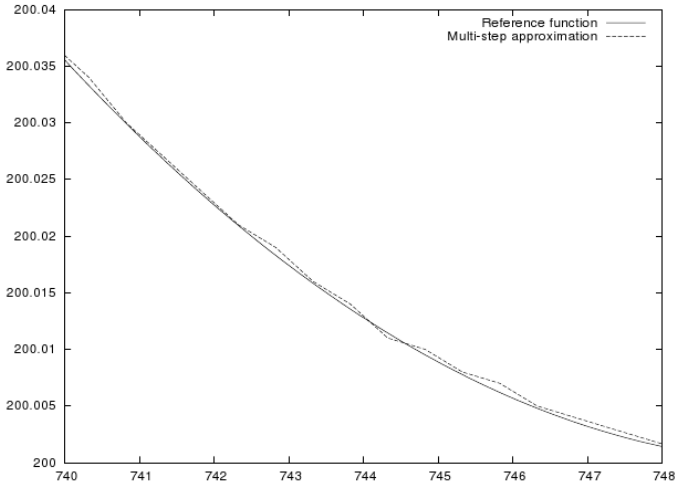


Figura 3.4: Comparación de la aproximación original y multi-step de una curva - acercamiento ($\Delta d = 2{,}5m$).

de la función. La posición actual sobre la función se obtiene estableciendo el próximo destino para el intruso. Luego pasamos al dominio del tiempo real en el cual el intruso se mueve de forma lineal a una velocidad fija s, establecida por el usuario.

En la figura 3.4 se muestra un ejemplo ilustrativo que representa un acercamiento de la función presentada anteriormente (ecuación 3.15). Como podemos observar, los diferentes segmentos permiten reconstruir la función original con un alto grado de precisión, siendo esta precisión ajustable variando el parámetro Δd referenciado anteriormente.

3.4. Descripción de herramienta generadora de eventos para WSNs

La herramienta desarrollada para generar eventos para WSNs integra dos clases representativas de eventos en tiempo-real, las cuales son rastreo y seguimiento de intrusos, y monitorización de la propagación de fuego o gas. Los eventos asociados con intrusos móviles están soportados, ofreciendo varios patrones de movilidad y opciones. Con respecto a los eventos de propagación de fuego y gas, se consideran escenarios tanto para interiores como para exteriores. El generador de eventos también soporta el formato TCL para la salida, que es completamente compatible con el conocido simulador ns-2, permitiendo así la integración con la plataforma de simulación.

La figura 3.5 muestra los principales componentes de la herramienta generadora de eventos para WSN. Como entrada de datos tenemos: (a) el tipo de evento (intruso, gas o fuego), donde cada tipo de evento puede tener una subcategoría; (b) el escenario,

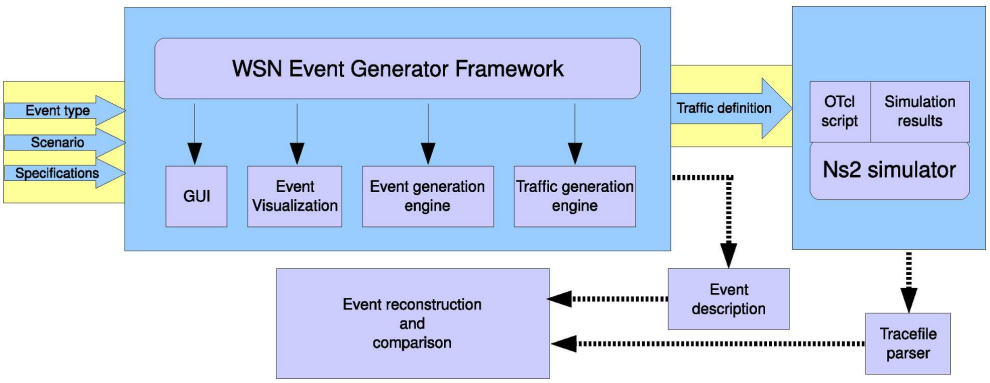


Figura 3.5: Arquitectura de la herramienta para generación y evaluación de eventos WSNs.

que deberá ser considerado bajo las especificaciones del estándar IEEE 802.15.4, y que está compuesto de un área objetivo y un despliegue de sensores estratégico; y finalmente, (c) la especificación de las condiciones del evento a modelar, tales como la tasa de propagación de gas, o la velocidad del viento y el ángulo, para los eventos de fuego.

Los módulos principales de esta herramienta son: (1) interfaz gráfica de usuario, (2) visualizador de eventos, (3) motor generador de eventos y (4) motor generador de tráfico.

La interfaz gráfica de usuario permite la interacción entre el usuario final y la herramienta propuesta. Esta interfaz permite proporcionar los datos de entrada, visualizar los eventos y generar el tráfico para esos eventos. También permite obtener una visualización gráfica de la reconstrucción de eventos y de las medidas de precisión.

El módulo de visualización de eventos permite ver el movimiento o el patrón de propagación para el evento usando gráficas directamente relacionadas con el escenario actual, y de acuerdo a las especificaciones proporcionadas por la herramienta de simulación.

El motor de generación de eventos es el módulo responsable de analizar cada uno de los tipos de eventos soportados por la herramienta con sus características particulares. Con este propósito, este módulo calcula el comportamiento a través del tiempo usando algoritmos de crecimiento o progreso.

El motor de generación de tráfico usa los resultados producidos por el motor generador de eventos para crear tráfico de acuerdo al formato especificado por el simulador de redes ns-2 [72]. Confiando en un sistema de detección binaria, el tráfico se inicia cuando los nodos sensores detectan el evento; en caso contrario permanecen inactivos. Los datos de entrada para el simulador ns-2 también definen el protocolo de encaminamiento usado, la capa MAC y el escenario. Tanto el tamaño y, la estrategia de despliegue de los nodos, como la posición inicial del dreno, se definen explícitamente. Finalmente, el módulo de reconstrucción de eventos combina los resultados de la

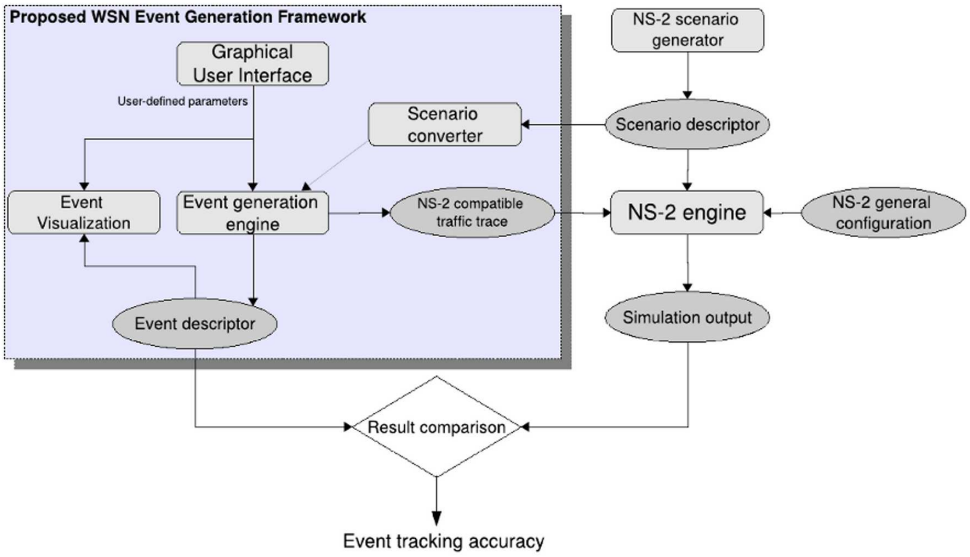


Figura 3.6: Componentes del generador de eventos WSN propuesto y la vinculación con el simulador ns-2.

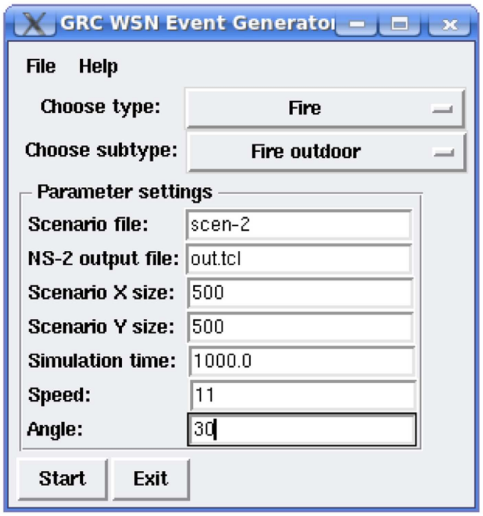
simulación con la descripción del evento original para reconstruir los eventos basados en el tráfico recibido en el dreno. Además, en este módulo se puede comparar el evento original con el reconstruido, funcionalidad que es proporcionada por las métricas de precisión.

La herramienta propuesta fue desarrollada utilizando un lenguaje interpretado (Perl/Tk) para hacer que esta fuera simple de usar y modificar, soportando tanto el funcionamiento en modo consola como interfaz gráfico.

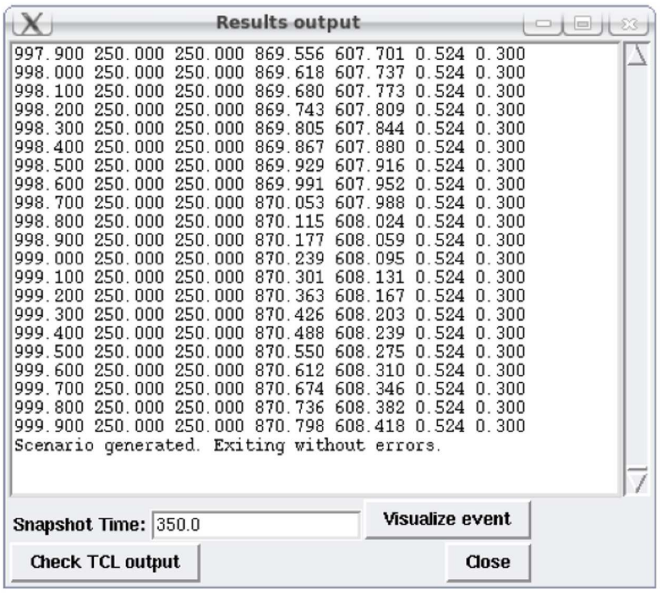
La figura 3.6 ofrece una visión general del generador de eventos WSN propuesto, la cual permite evaluar la precisión del proceso de rastreo de eventos a través del tiempo utilizando la información proporcionada por la WSN. Podemos observar que la herramienta propuesta integra varios componentes, usando los parámetros de eventos definidos por el usuario y la descripción del escenario como entrada. La salida consiste en un fichero *descriptor de eventos*, así como de un fichero de *tráfico de traza* compatible con ns-2.

Observe que, a pesar de que la herramienta para generar eventos WSN fue diseñada para ser compatible con ns-2, su proceso de desarrollo modular permite fácilmente adaptar su salida para que sea compatible con otros simuladores de WSN de interés. De hecho, los únicos componentes que se necesitan actualizar son el conversor de escenarios y la función específica en el motor del generador de eventos.

En la figura 3.7 se muestran las partes más representativas de la interfaz gráfica de usuario desarrollada, la cual representa un *front-end* para el motor generador de eventos. En la figura 3.7 a) se muestra el menú principal, donde los usuarios seleccionan



a)



b)

Figura 3.7: Ventana principal (a) y ventana de resultados de salida (b) del front-end propuesto para el framework generador de eventos WSN.

Algoritmo 3.3 Activación binaria de nodos del evento de acuerdo al modelo de propagación de gas.

```
while (t_i < simulation_end_time)
  for each n in nodes do
    if (||C − S_n|| < s · t_i)
      set node_ n ON
      update_ traffic
    end if
  end for each
end while
```

el tipo/subtipo de evento deseado, así como los parámetros requeridos para el evento específico. Pulsando *inicio* se hace la llamada al motor generador de eventos, y se muestra una ventana con los resultados tal y como se observa en la figura 3.7 b) Allí el usuario tiene todos los detalles relacionados con el evento, incluyendo la posibilidad de visualización de la salida de ns-2 (formato TCL), así como una visión gráfica del evento en cualquier instante del tiempo. Para esta última tarea se hace uso de la flexibilidad de generación de gráficas que ofrece la herramienta Gnuplot [73].

3.4.1. Integración de los algoritmos propuestos con la herramienta generadora de eventos

En cualquier aplicación de detección de eventos se debe considerar un punto inicial para cada evento, que es detectado al inicio por sólo unos pocos nodos sensores. Estos deberán enviar la información a través de los nodos vecinos hasta el dreno. Dependiendo del tipo de evento, se podrá mover o propagar a través del área monitorizada, provocando que más nodos detecten el evento. Estos deberán actuar de forma similar a los primeros nodos sensores que detectaron el evento, enviando la información a través de los nodos vecinos para llegar hasta el nodo dreno.

3.4.1.1. Algoritmo de expansión del gas

El patrón de propagación de gas en escenarios de interior es un proceso que depende de las condiciones volumétricas del ambiente. Para este tipo de eventos, en este capítulo usamos un modelo basado en un patrón de expansión en anillo, que garantiza la efectividad del proceso de rastreo del contorno basado en los datos recibidos en el dreno.

El algoritmo 3.3 explica de forma general el modelo usado para representar la expansión de gas y la respectiva activación de sensores en el evento.

La propagación de gas en escenarios internos es un proceso que es relativamente simple de modelar, dependiendo principalmente de las condiciones volumétricas del medio ambiente. Nuevamente el modelo está basado en el patrón de expansión en anillo, el cual es adecuado para pruebas objetivas en las WSN. Con este modelo, el usuario simplemente proporciona la posición inicial *(h, k)* representada como C en el

Algoritmo 3.4 Activación binaria de nodos del evento de acuerdo al modelo de propagación de fuego.

```
while (t_i < simulation_end_time)
  for each n in nodes do
    if (||F_1^t − S_n|| + ||F_2^t − S_n|| ≤ 2 · a_t · (1 + λ))
      set node_n ON
      update_traffic
    end if
    if (||F_1^t − S_n|| + ||F_2^t − S_n|| ≤ 2 · a_t)
      set node_n OFF
      destroy node_n
      update_traffic
    end if
  end for each
end while
```

algoritmo, la velocidad de propagación (s) para el radio del círculo R_g, el cual define el perímetro del gas y su incremento, y la posición de cada sensor (S_n).

3.4.1.2. Algoritmo de expansión del fuego

El patrón de propagación del fuego en ambientes externos depende principalmente de la velocidad y dirección del viento. Adicionalmente, la propagación del fuego también depende del tipo de material que se esté quemando. Se propone un modelo de propagación de fuego forestal usando la expansión en elipse, donde uno de los puntos se fija en el origen del fuego. El algoritmo 3.4 muestra el pseudocódigo usado para este evento.

En el algoritmo, a_t se define como:

$$a_t = r_w \cdot t \cdot \left[\frac{1 + \left(\frac{r_0}{r_w}\right)^2}{2} \right]$$

Los sensores se activan cuando están físicamente dentro del área de fuego modelada. Si un sensor se localiza en el punto S_n y los dos puntos de la elipse que se usan para representar el área afectada en el tiempo t se encuentran en las coordenadas F_1^t y F_2^t, el sensor será activado si se satisface la condición mostrada en el algoritmo 3.4.

Como se puede ver, solo los sensores cuya suma de las distancias ó los focos sea superior a $2 \cdot a_t$ pero inferior a $2 \cdot a_t \cdot (1 + \lambda)$ enviarán tráfico al dreno.

3.4.1.3. Integración de eventos basados en intrusos en la herramienta generadora de eventos

En la sección 3.3 se ofrece una descripción teórica de varios patrones de movilidad de intrusos. Los diferentes patrones de movilidad propuestos fueron integrados en nuestra herramienta para generar eventos en tiempo real. Este proceso necesita tener

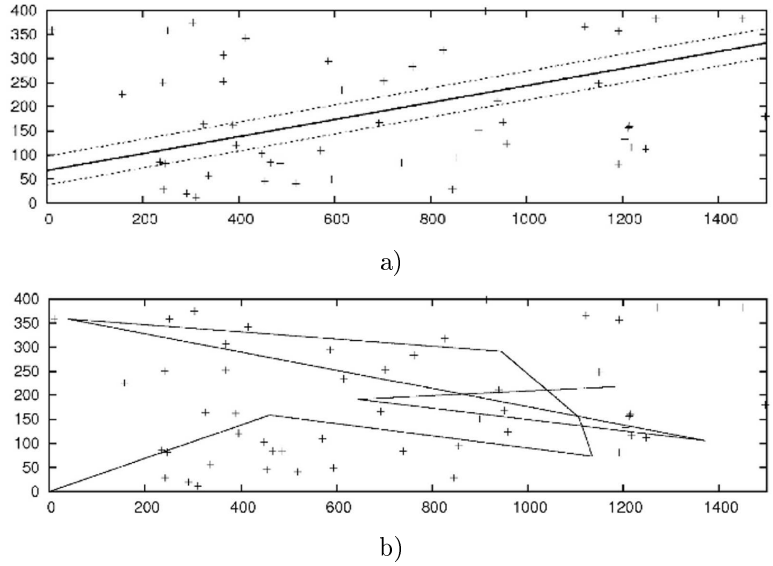


Figura 3.8: Movimiento del intruso de acuerdo al modelo de movimiento recto (a) y modelo *random waypoint* (b).

como entrada una traza de las posiciones de los sensores en un formato compatible con ns-2. Siguiendo un análisis de tiempo discreto, combinamos información relativa a la posición de los sensores con la posición del intruso en un cierto instante de tiempo.

Los sensores se activan cuando la distancia entre el intruso y los sensores está por debajo de un umbral mínimo (δ) necesario para que este último sea activado. Este valor es configurado por el usuario, y depende de la tecnología y algoritmo utilizado.

La integración de este proceso en nuestra herramienta requiere la comprobación, para cada sensor localizado en un punto S_i, si la condición siguiente se cumple:

$$\|S_i - P_t\| \leq \delta \tag{3.17}$$

donde P_t representa la posición del intruso en el tiempo t.

En la figura 3.8 se presentan las gráficas de diferentes patrones de movilidad de intrusos generadas por la herramienta. En la figura 3.8 (a) se muestra la ruta seguida por un intruso cuando su movimiento es en línea recta. Solo para este caso incluimos dos líneas punteadas, las cuales definen el límite del área para disparar los sensores, significando que los sensores (representados como cruces) dentro del área serán activados cuando el intruso esté cerca.

La figura 3.8 (b) muestra la ruta que sigue un intruso cuando se mueve de acuerdo al algoritmo *random waypoint*. En este caso, el intruso entra al escenario desde la parte inferior izquierda y se mueve aleatoriamente hacia los *Puntos de Interés* (POIs) siguiendo trayectorias rectas.

Otras posibles aplicaciones del modelo genérico desarrollado se presentan en la

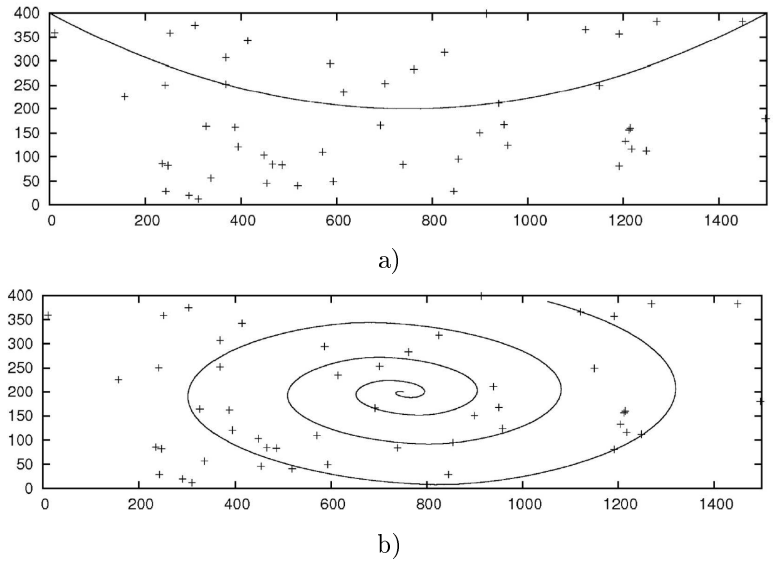


Figura 3.9: Movimiento del intruso de acuerdo a una curva (a) y una espiral (b) usando el modelo genérico.

figura 3.9. En la figura 3.9 a) se muestra la ruta seguida por un intruso, el cual entra al escenario por la parte superior izquierda y después se mueve de acuerdo a la curva definida por un polinomio de segundo orden. En la figura 3.9 b) se muestra una aplicación mas sofisticada del modelo genérico, donde el intruso inicia su movimiento en el centro del escenario y después se mueve de acuerdo a una espiral plana hasta que sale del área monitorizada.

3.5. Algoritmos de agregación de datos y reconstrucción de eventos

La detección de propagación de eventos como gas y fuego involucra varios sensores distribuidos dentro del área de interés. Un despliegue de nodos con alta densidad permite más precisión en el rastreo de eventos, aunque en la práctica la distancia entre los sensores se define normalmente siguiendo los límites del estándar de comunicación inalámbrica utilizado (IEEE 802.15.4 en nuestro caso).

Cuando hablamos de eventos que se propagan a través de un escenario es deseable permitir una visualización de la monitorización de los límites de propagación. En estos casos, los límites pueden ser definidos por un polígono de dos dimensiones que conecta a los nodos sensores, detectando el evento cerca del perímetro de la región afectada [74]. Por otra parte, el seguimiento de intrusos en tiempo-real requiere una reconstrucción de la trayectoria del intruso. En las siguientes subsecciones se presentan

los algoritmos de cada una de las categorías de eventos analizadas y evaluadas en este capítulo.

3.5.1. Algoritmo propuesto para la reconstrucción de eventos de gas y fuego

Nuestra propuesta busca una alta efectividad en el proceso de rastreo del contorno de las áreas afectadas por gases o fuego basándose en los datos recibidos en el dreno. Los algoritmos formales utilizados para reconstruir los eventos de gas y fuego mediante el proceso de calcular las diferencias (error) entre el evento original y el reconstruido se describen a continuación. El algoritmo 3.5 muestra el proceso para la reconstrucción de eventos de gas, mientras que el algoritmo 3.6 describe el proceso de reconstrucción para el fuego. Ambos algoritmos son básicamente iguales, solo difiriendo en lo que respecta al evento original utilizado como referencia.

En el algoritmo de reconstrucción de eventos de gas primero se obtienen los datos para el instante de tiempo deseado mediante el primer ciclo *while (line = <event_file>),* almacenando los parámetros del evento de referencia en las variables *posx, posy y posz.* Los dos ciclos *for* siguientes permiten llenar el área que corresponde al evento de referencia, haciendo un recorrido del total de la dimensión tanto en x como en y. Después se ejecutan las funciones *read_sensores_scenario()* para obtener la posición de los sensores desde el escenario de ns-2 y a la función *get_sensor_activity()* para obtener, desde el fichero de traza de ns-2, el historial de activación de los sensores hasta el instante en el que se desea realizar la reconstrucción. En los siguientes ciclos *for* se conectan entre ellos los sensores activos que estén cercanos entre sí. Finalmente, las funciones *get_border()* y *fillin_area()* nos muestran gráficamente el área estimada de propagación de gas para el periodo de tiempo analizado.

En el algoritmo de reconstrucción de eventos de fuego es un proceso muy similar al del gas, cambiando básicamente la parte que permite definir el perímetro de referencia debido a la ecuación que parametriza el comportamiento del fuego, que es diferente a la del comportamiento del gas.

El cálculo del error es doble: por un lado medimos el promedio del error del borde calculando la distancia media entre el perímetro real y el perímetro estimado, y, por otro lado, medimos el error del área comparando el área real contra la estimada.

Como ejemplo de las medidas de error obtenidas en cada uno de los eventos, mostramos el proceso de reconstrucción para el evento del fuego en un instante de tiempo específico. La figura 3.10 (a) muestra como los diferentes nodos sensores activos están conectados por el algoritmo de reconstrucción para crear una distribución en malla (*grid*). La figura 3.10 (b) muestra el área estimada y el área de referencia para calcular el grado de solape. La figura 3.10 (c) muestra el borde estimado para el evento, y la figura 3.10 (d) muestra la estimación del error en 360 grados.

Con estos resultados obtenemos básicamente dos métricas: (1) el área de error, presente entre el área reconstruida y el área real (solo en la sección de solapamiento), y (2) el error del borde, medido como el promedio del error obtenido en 360 grados. Por ejemplo, considerando la figura 3.10 (d), este sería el promedio de los valores representados.

Algoritmo 3.5 Proceso general de reconstrucción de eventos de gas.

```
#obtain reference data for desired time_step
while (line = <event_file>)
      if (found == 0) then
           #print "Line is: line";
           set vals = split(" ",line);
           if (vals[0] >= ARGV[3]) then
                set found = 1;
                set posx = vals[1];
                set posy = vals[2];
                set posz = vals[3];
                set radius = vals[4];
           end if
      end if
end while
# fill-in reference array
set num_values=0;
for each i=0 to size_x do
      for each j=0 to size_y do
           set tmp1 = (i-$posx)**2 + (j-posy)**2;
           if (tmp1 <= radius**2) then
                set arr_ref[i][j]=1;
                set print ref_file "i j \n";
                set num_values++;
           end if
      end for
end for
# obtain sensor positions from ns-2 TCL scenario data
call read_sensor_scenario();
#obtain the sensors activity from ns-2 traffic file
call get_sensor_activity();
set num_values=0;
for each node=0 to num_nodes do
      if (act_nodes[node] == 1) then
           for each node_next = node+1 to num_nodes do
                if (act_nodes[node_next] == 1) then
                   #connect node a node_next
                   set ret = connect_nodes(node, node_next);
                   if (ret) then
                      set num_values++
                   end if
                end if
           end for
      end if
end for
call get_border();
call fillin_area();
```

Algoritmo 3.6 Proceso general de reconstrucción de eventos de fuego.

```
while (line = <event_file>) do
      if (found == 0) then
           #print "Line is: line";
           set vals = split(" ",line);
           if (vals[0] >= ARGV[3]) then
                set found = 1;
                set val_a = vals[1];
                set val_b = vals[2];
                set val_h = vals[3];
                set val_k = vals[4];
                set val_angle = vals[5];
           end do
      end do
end while
# fill-in reference array
set num_values = 0;
set val_c = sqrt (val_a**2 - val_b**2);
set val_F1_x = val_h - val_c * cos(val_angle);
set val_F1_y = val_k - val_c * sin(val_angle);
set val_F2_x = val_h + $val_c * cos(val_angle);
set val_F2_y = val_k + val_c * sin(val_angle);
for each i=0 to size_x  do
      for each  j=0 to size_y do
           set dist_F1 = sqrt((i-val_F1_x)**2 + (j-val_F1_y)**2);
           set dist_F2 = sqrt((i-val_F2_x)**2 + (j-val_F2_y)**2);
           if (dist_F1 + dist_F2 <= 2*val_a) then
                set arr_ref[i][j]=1;
                print reffile "i j\n";
                set num_values++;
           end if
      end for
end for
# obtain sensor positions from ns-2 TCL scenario data
call read_sensor_scenario();
call get_sensor_activity();
# fill-in simulation array
set num_values=0;
for each node=0 to num_nodes do
      if (act_nodes[node] == 1) then
           for each node_next = node+1 to num_nodes do
                if (act_nodes[node_next] == 1) then
                   #connect node a node_next
                   set ret = connect_nodes(node, node_next);
                   if (ret) then
                      num_values++
                   end if
                end if
           end for
      end if
end for
call get_border();
call fillin_area();
```

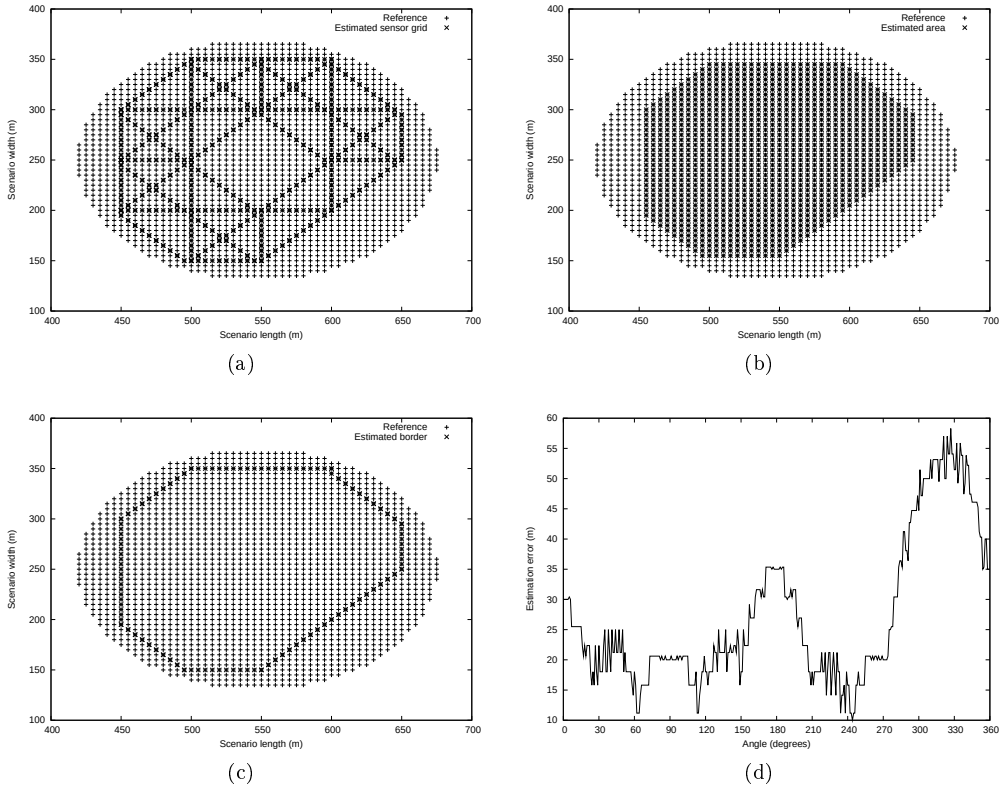


Figura 3.10: Representación gráfica de la estimación del error en los eventos de gas y fuego en un instante de tiempo específico.

3.5.2. Algoritmo propuesto para el seguimiento de intrusos en tiempo-real

Los sistemas de tiempo real requieren una comunicación confiable y retardos pequeños. Debido a que las aplicaciones de WSNs para detección y rastreo de intrusos deben trabajar de manera cercana al tiempo real, también tienen como objetivo valores de retardo bajo. Dentro del alcance del rastreo de intrusos, el reto es asegurar que los retardos desde diferentes fuentes sean lo suficientemente pequeños como para permitir la agregación oportuna de datos, y además para obtener una estimación de la posición del intruso tan precisa como sea posible. Para cumplir este objetivo eficientemente deberemos disponer de: (a) algoritmos de encaminamiento que sean robustos y resistentes a los cambios de topología, así como la respuesta apropiada a las restricciones del sistema de sensores inalámbricos implementada, imponiendo al mismo tiempo una baja sobrecarga de encaminamiento; (b) algoritmos de agregación de datos que sean capaces de rastrear los intrusos usando un número pequeño de mensajes, evitando el uso de grandes cantidades de tiempo para la recopilación de informes, y también siendo flexibles en la presencia de pérdida de paquetes.

En nuestro enfoque, los siguientes aspectos deberán ser considerados:

- La detección de sensores es binaria con un rango de 10 metros (igual a la mínima distancia entre sensores), y la cobertura de los sensores es omnidireccional. Esto implica que, para la estrategia de despliegue propuesta (en malla), el número de sensores que detectan un intruso puede estar en el rango de uno a cuatro.
- La posición del intruso es constantemente actualizada por el nodo dreno en base a la información enviada por los diferentes sensores, y tomando en cuenta el tiempo de llegada asociado con cada uno.

El algoritmo 3.7 resume el proceso utilizado para la estimación de la posición de intrusos, el cual está basado en los mensajes recibidos de los sensores. Este algoritmo adopta una estrategia para la agregación de los datos capaz de estimar la posición de un intruso en movimiento rápidamente y con un buen grado de precisión, a pesar de que la detección realizada por los sensores es binaria. Las siguientes operaciones son llevadas a cabo:

- Cada grupo agrega mensajes de varios sensores, combinando esos datos para obtener una estimación de la posición del intruso. Grupos consecutivos estarán separados por al menos un tiempo especificado en segundos (*interval*) definido por el usuario.
- El vector $\overrightarrow{P_e}$ contiene la posición estimada del intruso, y éste es constantemente actualizado con base en la posición asociada con los diferentes sensores ($\overrightarrow{P_s}$) que detectan el intruso. Cada mensaje tiene asociado un identificador único (*id*) por el dreno.

Algoritmo 3.7 Proceso de estimación de la posición de intrusos.

```
input: interval, max_alfa, beta
begin
set microint = interval/5; #microgroup interval
set last_id = 0;
set \overrightarrow{P_e}[0] =
\overrightarrow{P_s}[0]; #initial position equal to the position of the first sensor reporting
for each report id received do {

    if (timestamp[id] - timestamp[0] < $interval) {   # for first group only

        set alfa = \frac{timestamp[id]-timestamp[id-1]}{timestamp[id]-timestamp[0]} \times beta

        if (alfa > max_alfa) set alfa = max_alfa;

        set \overrightarrow{P_e}[id] = \alpha \cdot \overrightarrow{P_s}[id] + (1-\alpha) \cdot \overrightarrow{P_e}[id-1]

    } else {   # group-based estimation
       if (timestamp[id]-timestamp[last_id+1] < microint) { # microgroup detected

            set \overrightarrow{P_{mgr}} = average(\overrightarrow{P_s}[last\_id +
            1] to \overrightarrow{P_s}[id])   #microgroup estimation

        } else {     # new microgroup

            set \overrightarrow{P_{mgr}} =
            \overrightarrow{P_s}[id]; #est. group pos. equal to current sensor pos.
            set last_id = id - 1;
        }

        set \overrightarrow{V_{mgr}} = estimate_intruder_speed(last_id, id) #reports from last_id to id

        set \overrightarrow{P_{speed}} =
        \overrightarrow{P_e}[last\_id] + \overrightarrow{V_{mgr}} \times (timestamp[id] - timestamp[last\_id])

        set alfa = \frac{timestamp[id]-timestamp[last\_id]}{microint} \times beta

        if (alfa > max_alfa) set alfa = max_alfa;

        set \overrightarrow{P_e}[id] = \alpha \cdot \overrightarrow{P_{mgr}} + (1-\alpha) \cdot \overrightarrow{P_{speed}}   }

} end
```

- En el primer grupo se estima la posición del intruso mediante un filtro exponencial que utiliza valores nuevos y anteriores. El factor α caracteriza el comportamiento de esta estimación: valores más altos hacen al sistema más sensible a cambios drásticos en la señal, mientras que valores más bajos hacen a este más conservador.

- A partir del segundo grupo en adelante, todos los grupos que monitorizan están divididos en micro-grupos, y la estimación de la posición ($\overrightarrow{P_{mgr}}$) y la velocidad ($\overrightarrow{V_{mgr}}$) se llevan a cabo por cada micro-grupo. La posición del intruso estimada se basa nuevamente en un filtro exponencial que utiliza la estimación de la posición de los micro-grupos ($\overrightarrow{P_{mgr}}$) y la estimación de la posición derivada de la velocidad ($\overrightarrow{P_{speed}}$). El último se calcula como la proyección de la posición anterior estimada ($\overrightarrow{P_e}[last\_id-1]$), mas la distancia proporcionada por el vector velocidad para el intervalo de tiempo entre los dos micro-grupos.

- El parámetro α se calcula dinámicamente, incrementándose como la diferencia del tiempo entre el tiempo actual y el incremento del tiempo inicial del microgrupo. Los parámetros ($\beta\epsilon[0,1]$) se utilizan para controlar el crecimiento de α, los cuales nunca se incrementan más allá de su límite superior ($max\_alfa\ \epsilon$ [0,1]).

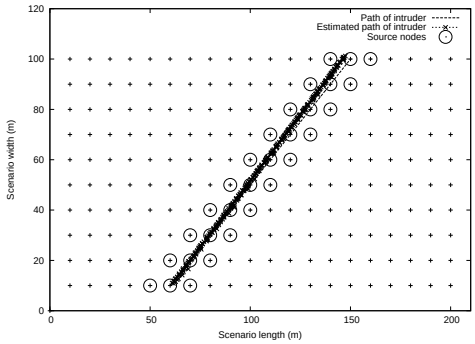
Finalmente, la salida de este algoritmo consiste de la secuencia de posiciones estimadas del intruso ($\overrightarrow{P_e}$), las cuales son relacionadas con el tiempo correspondiente para comparaciones contra la secuencia real de posiciones del intruso al final.

Con el propósito de verificar el funcionamiento correcto del algoritmo propuesto y para garantizar la exactitud del rastreo, se propone un conjunto de patrones de movilidad de intrusos (linea recta, aleatoria y curva), como se muestra en la figura 3.11. En estas figuras podemos ver, además de las rutas del intruso representadas como una trayectoria de línea punteada, algunos círculos que resaltan los nodos sensores que están activos en algún instante de tiempo, generando el mensaje apropiado para el dreno. La secuencia de cruces representa la secuencia de estimaciones acerca de la ruta del intruso hecha por el dreno, y basada en los diferentes mensajes recibidos. Las diferencias entre las trayectorias real y estimada del intruso son más notorias en la figura 3.11.b), aunque el tiempo de variación asociado con la estimación de retardos no se pueda observar en la misma.

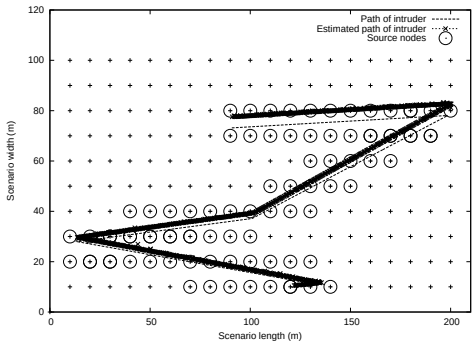
Para obtener la medida de precisión del error total en determinadas trayectorias de intrusos, se llevan a cabo los siguientes pasos:

1. Recuperar la ubicación exacta del intruso en todos los instantes de tiempo basados en el patrón de movilidad definida como entrada a la simulación.

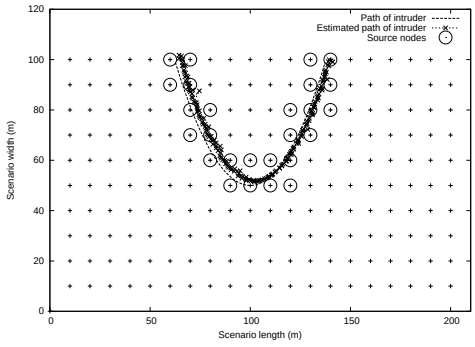
2. En el dreno se lleva a cabo la recepción de mensajes recibidos por parte de los sensores que detectaron al intruso. Los datos disponibles son: número de sensores que detectan al intruso, la posición de cada sensor, y el instante de tiempo del mensaje de la advertencia del intruso por cada sensor.



a)



b)



c)

Figure 3.11: Ejemplos de la precisión del seguimiento de intrusos para diferentes patrones de movilidad: a) línea recta, b) aleatorio y c) curva.

3. Por cada mensaje recibido, se obtiene una nueva posición estimada para el intruso de acuerdo con el algoritmo 3.7, el cual crea grupos de información combinando los mensajes recibidos en intervalos de tiempo de duración fija.

4. Finalmente se calcula la distancia euclidiana media entre la secuencia de posiciones del intruso conocidas y las estimaciones de la trayectoria del intruso.

Para nuestras pruebas, los valores adoptados para los diferentes parámetros del algoritmo son: $interval = 5$ s, $max\_alfa = 0{,}25$, y $\beta = 0{,}4$.

Dentro del error calculado, tres factores diferentes se combinan: (a) la posición imprecisa del intruso asociada con el reporte binario proporcionada por los sensores; el error podrá ser igual a 10 metros para el primer mensaje recibido ya que éste es el rango de detección del sensor, (b) el error medio estimado introducido por el algoritmo de agregación de datos elegido, a partir de la información recibida; y (c) el retardo experimentado por los diferentes mensajes cuando viajan a través de la WSN hacia el dreno.

3.6. Sumario

Las redes de sensores inalámbricas (WSNs) diseñadas para tareas críticas deberán ser capaces de ofrecer retroalimentación acerca de los eventos, permitiendo supervisar estos eventos con un grado de precisión razonable dentro de estrictos límites de retardo.

Lograr una respuesta en tiempo-real requiere varias mejoras a los protocolos y tecnologías actualmente disponibles para WSN en diferentes capas; tales mejoras típicamente se basan en simulaciones, al menos en una fase preliminar. Para evaluar con precisión la aplicación de las WSNs en el rastreo de eventos en tiempo-real, se desarrolló un generador de eventos compatible con el simulador ns-2 que es capaz de modelar tanto eventos para detección de intrusos, como eventos de propagación de fuego o gas en escenarios de interior o exterior. En este capítulo se ha presentado una descripción analítica de la herramienta desarrollada, y también se presentaron ejemplos visuales de diferentes tipos de eventos. Para cada tipo de evento que se puede generar con la herramienta propuesta, se ha propuesto un algoritmo específico que pretende simular los principales aspectos que intervienen en cada uno de dichos eventos de forma real. Los resultados obtenidos en la simulación, relativos al desempeño de la red y de los algoritmos de reconstrucción de eventos, serán presentados en el capítulo 5.

Capítulo 4

Encaminamiento eficiente en WSNs con drenos estáticos y móviles

El objetivo de los protocolos de encaminamiento es proporcionar una comunicación fiable y eficiente entre los nodos de una red. En este capítulo se proponen dos protocolos de encaminamiento para WSNs con drenos estáticos y móviles. El primero de ellos se denomina DABR (*Drain Announcements Based Routing*), y está basado en anuncios periódicos del dreno, siendo idóneo para entornos donde los sensores y el dreno son estáticos. El segundo protocolo propuesto es el MRLG (*Mobile-drain Routing for Large Grids*), el cual está orientado a redes en las cuales el nodo dreno es móvil, y por lo tanto constantemente se están actualizando las rutas de los nodos hacia el dreno.

4.1. Introducción

Las prestaciones de las WSNs en la monitorización de eventos de tiempo crítico es una preocupación importante, principalmente debido a la necesidad de garantizar que las acciones a ejecutar como respuesta a estos eventos sean las adecuadas. Utilizando el estándar IEEE 802.15.4 para la monitorización de eventos de tiempo crítico en WSNs, hemos desarrollado un esquema de encaminamiento basado en anuncio del dreno llamado DABR [75], y otro llamado MRLG [76] para redes con drenos móviles, teniendo ambos la finalidad de minimizar la sobrecarga de encaminamiento.

4.2. DABR: Esquema de encaminamiento basado en anuncio del dreno para WSNs

Con este protocolo de encaminamiento se pretende reducir la sobrecarga de encaminamiento para el descubrimiento de rutas por los nodos sensores, cuando requieren enviar información al dreno de la red. El algoritmo propuesto también pretende reducir el retardo extremo a extremo al generar poco tráfico de encaminamiento en los canales de comunicación. La implementación de este protocolo de encaminamiento se basa en anuncio del dreno, y está orientado a escenarios en los que los nodos sensores y el nodo dreno son estáticos, distribuidos en una topología de malla dentro de un espacio físico. El protocolo de encaminamiento envía mensajes *broadcast* para descubrir rutas, permitiendo que los nodos sensores sean localizados y asociados con el nodo dreno. La ruta se mantiene actualizada aún en casos en los que algunos nodos sensores no estén habilitados o pierdan la ruta en sus respectivas tablas de encaminamiento. El protocolo contempla que, en intervalos periódicos, se anuncie la presencia del dreno, para mantener actualizadas las tablas de encaminamiento.

Para explicar el esquema de encaminamiento basado en anuncio del dreno, mencionaremos primero que este esquema de encaminamiento está enfocado a WSNs multi-salto con un solo dreno, donde los nodos sensores están distribuidos con una densidad elevada. Normalmente se optimiza el despliegue de los sensores buscando que la distancia entre ellos esté determinada por el rango de transmisión de la tecnología IEEE 802.15.4 (10 metros). En el caso del nodo dreno, este es un nodo con una posición fija que recibe la información enviada desde diferentes nodos sensores. El despliegue de los nodos en estas aplicaciones puede ser aleatorio o manual, haciendo notar que la ubicación de los nodos sensores y la distancia entre ellos será importante para el desempeño del protocolo de encaminamiento.

A continuación se presenta el esquema de encaminamiento basado en anuncio del dreno:

1. El nodo dreno se anuncia mediante mensajes *broadcast* con números de secuencia incrementales.

2. Los nodos vecinos al dreno (nodos dentro del rango de transmisión del dreno) reciben el anuncio del dreno, por lo que almacenan la ruta hacia el nodo fuente (dreno) y envían mensajes *broadcast* a todos los nodos vecinos.

3. Si un nodo recibe un mensaje que contiene una ruta hacia el nodo dreno más de una vez, este da preferencia al que tiene mayor número de secuencia; si el número de secuencia es el mismo analiza también que el número de saltos sea menor del que tiene almacenado para llegar al dreno.

4. Cada entrada en la tabla de encaminamiento tiene asociado un tiempo de vida límite, durante el cual la ruta será válida.

5. Las rutas se actualizan a través de anuncios del dreno periódicos, que son propagadas hacia todos los nodos.

Algoritmo 4.1 Generación de mensajes anunciando al dreno.

```
Input: Sink_ID, interval, stop_time
Variables: packet, time, broadcast_id

BEGIN
  time = 0
  broadcast_id = 0
  REPEAT
    DrainNotify(Node_sink)
    time += interval
  UNTIL (time < stop_time)
END

FUNCTION DrainNotify(Node_source)
  VAR broadcast_id, request, rtable, packet
  packet.Node_source = Sink_ID
  packet.Node_dst = broadcast_addr
  packet.msg_type = DRAIN_ANNOUNCEMENT
  packet.msg_seqnum = broadcast_id++
  packet.hop_count = 0
  broadcast(packet)
END DrainNotify
```

6. La información de la ruta se mantiene en la tabla de encaminamiento de cada nodo hasta que el enlace con los nodos vecinos se pierde, o hasta que el tiempo de vida llega a su límite.

4.2.1. Descripción formal del protocolo DABR

El algoritmo 4.1 muestra la representación algorítmica del descubrimiento de rutas adoptado por el esquema de encaminamiento propuesto. Con este algoritmo se explica de forma general la manera en que funciona este protocolo. En primer lugar, puede observarse que los parámetros de entrada son el identificador del modo dreno *(Sink_ID),* el intervalo para el envío de los mensajes *broadcast* y para actualizar la ruta en todos los nodos sensores *(interval),* y el tiempo de simulación *(stop_time).* Dentro del cuerpo principal del algoritmo tenemos un bucle para invocar a la función *DrainNotify.* Con esta función el dreno realiza el descubrimiento de ruta a través del envío de mensajes *broadcast.* El final del bucle se alcanza cuando el tiempo de ejecución es igual al parámetro *stop_time.* Por otro lado, la función *DrainNotify* permite al dreno enviar mensajes *broadcast.* Estos paquetes serán recibidos por todos los nodos sensores, actualizando su ruta hacia el nodo dreno.

A partir del mecanismo de encaminamiento descrito en el algoritmo, se podrán crear y mantener actualizadas las rutas de tal manera que cualquier nodo podrá enviar paquetes hacia el dreno usando el procedimiento estándar: el nodo consulta su tabla de encaminamiento para verificar si tiene una ruta valida hacia el dreno, y después envía la información usando dicha ruta. En caso de que no se tenga ninguna ruta

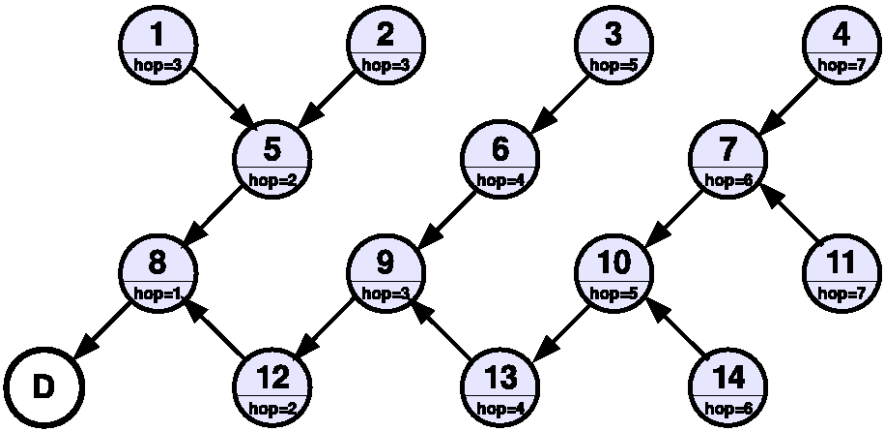


Figura 4.1: Escenario con 14 nodos y un dreno móvil.

disponible, todo el tráfico se descarta hasta que la ruta sea restaurada. Observamos que, cuando se realizan tareas de monitorización de eventos críticos, se espera que la retroalimentación de las redes sea cercana al tiempo real, y de esta manera los datos no permanecerán por largos periodos de tiempo en el *buffer*, con el riesgo de que se pierdan. Los datos recolectados por los nodos serán enviados al dreno a través de la ruta almacenada en cada uno de los nodos. Tan solo aquellos que pierdan su ruta por algún tipo de fallo tendrán que esperar un periodo de tiempo establecido por la variable *interval*, para actualizar nuevamente la ruta, considerando que durante este tiempo el *buffer* podrá mantener almacenada la información recolectada.

4.2.2. Limitaciones del protocolo DABR

En un escenario con varios nodos y un dreno móvil, en donde el dreno es el responsable de actualizar las rutas en todos los nodos sensores, la carga de encaminamiento puede variar principalmente debido a la movilidad que tenga el dreno en la red, de tal manera que, al tener que actualizar las rutas con mucha frecuencia, se generará mayor sobrecarga de encaminamiento en la red.

La figura 4.1 muestra un ejemplo de un escenario con 14 nodos sensores y un dreno móvil identificado como *D*. Cada uno de los nodos sensores tiene una entrada de la tabla de encaminamiento, donde almacena el número de saltos que necesita para llegar al dreno. Cuando el dreno cambia de ubicación, el único nodo vecino que tenía comunicación directa con el dreno (nodo 8) pierde el enlace, por lo que el protocolo de encaminamiento deberá actualizar la ruta de todos los nodos sensores, generando una alta sobrecarga de encaminamiento.

En la figura 4.2 se muestra para un determinado desplazamiento del dreno, los nodos que han modificado su ruta hacia el mismo. Con el movimiento realizado por el dreno, los nodos afectados que deben actualizar su número de saltos para llegar al dreno y/o su nodo siguiente son cinco (nodos identificados con números 1, 5, 8, 9 y

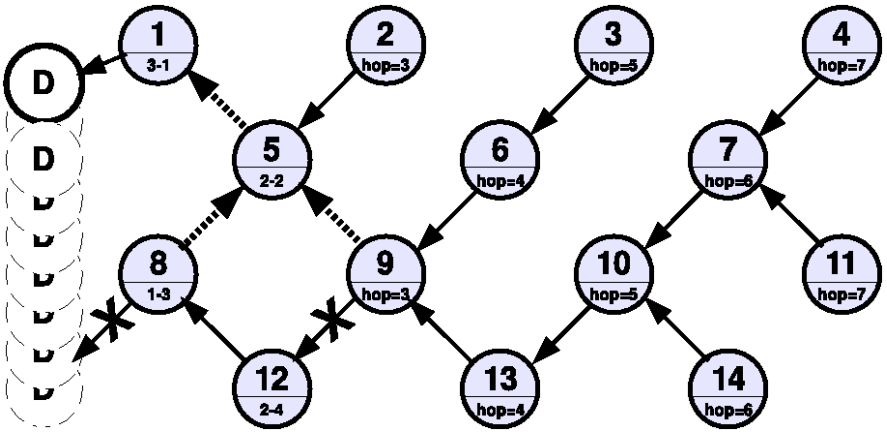


Figura 4.2: Encaminamiento básico con desplazamiento del dreno.

12), mientras que el resto de los nodos no modifica sus rutas.

4.3. MRLG: protocolo de encaminamiento con soporte para drenos móviles

En esta sección se presenta el algoritmo de encaminamiento MRLG, el cual está basado en el envío de mensajes por parte del dreno, que puede tener movilidad. En este protocolo, el dreno anuncia su presencia enviando mensajes *broadcast*. Los nodos que detecten su presencia pueden comprobar a cuántos saltos se encontraban del dreno previamente, de tal manera que actualicen su tabla de encaminamiento y propaguen la nueva posición del dreno. El protocolo tiene como objetivo minimizar la sobrecarga generada por las actualizaciones de las tablas de encaminamiento.

4.3.1. Funcionamiento del protocolo MRLG

Una vez realizado un análisis exhaustivo de las acciones necesarias para mantener actualizadas las tablas de encaminamiento debido al movimiento del dreno, se propone el nuevo protocolo MRLG para redes WSNs.

En la figura 4.3 se muestra de forma gráfica el funcionamiento del protocolo MRLG (continuando con el mismo escenario mostrado en la figura 4.2). Cuando el dreno se está moviendo a una nueva posición, también está enviando mensajes *broadcast*, que podrán recibir los nodos que se encuentren a menos de 10 metros de distancia. Cuando los mensajes son recibidos por el nodo 1, el cual tiene en su contador de saltos (*hop_count*) el valor de tres saltos, deberá actualizar este valor con el nuevo valor que es uno (está mejorando su *hop_count*) y deberá propagar el broadcast con la nueva ubicación del dreno. El nodo 5 sigue teniendo el mismo valor en *hop_count*, pero cambia el valor de su parámetro *next_hop*, que es propagado utilizando mensajes

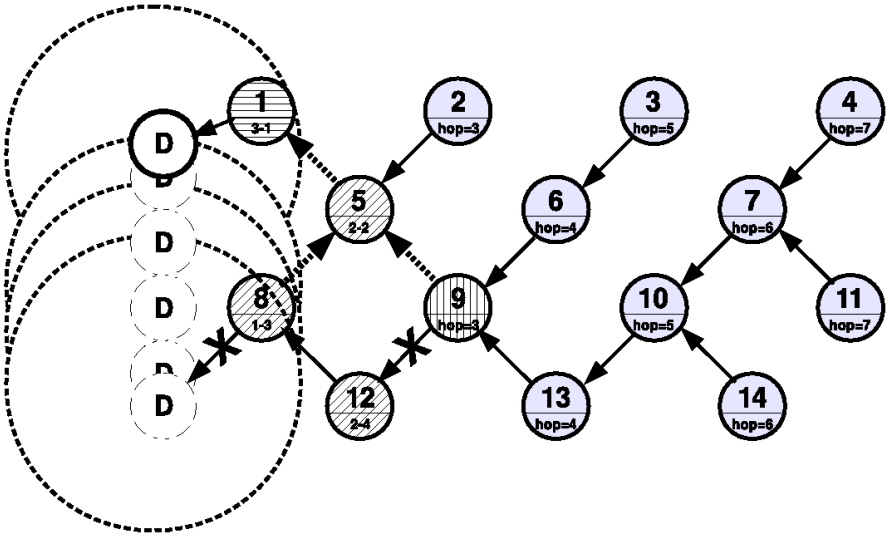


Figura 4.3: Funcionamiento del protocolo de encaminamiento MRLG.

broadcast. El nodo 2 no modifica los valores de los parámetros *hop_count* y *next_hop*, y por lo tanto no propaga la información. El nodo 8 empeora su *hop_count* y cambia su *next_hop*, por lo tanto propaga estos valores con mensajes *broadcast*. El nodo 9 sigue teniendo el mismo valor de *hop_count*, pero cambia su *next_hop*, por lo tanto los propaga. Por último, el nodo 12 se quedará con el enlace hacia el nodo 8, pese a haber recibido mayor *num_seq* de este nodo, porque se encontraba a la misma cantidad de saltos por la ruta del nodo 9. El resto de nodos permanece sin cambios, debido a que no les llegó ningún mensaje de control, reduciéndose así notablemente el número de mensajes enviados.

4.3.2. *Descripción formal del algoritmo MRLG*

El protocolo de encaminamiento ha sido optimizado para operar bajo las siguientes suposiciones, las cuales son aplicables al escenario objetivo: a) el número de nodos sensores no se incrementa en el tiempo, b) los nodos sensores se mantienen en posiciones estáticas, y c) el dreno es capaz de moverse libremente a través del área monitorizada, sin restricciones.

El protocolo MRLG distingue entre tres tipos de nodos vecinos, desde la perspectiva de un nodo sensor en particular: 1) *nodos downhill*: incluye a los nodos que están más cerca del dreno (contadores de saltos más bajos); 2) *nodos peers*: incluye los nodos a la misma distancia del dreno (similar contador de saltos) y 3) *nodos uphill*: incluye los nodos más retirados del dreno (contador de saltos más alto).

La secuencia de acciones adoptadas por el algoritmo de encaminamiento MRLG cuando se actualizan las rutas es la siguiente:

Algorithm 4.2 Actualización de rutas: propagación condicional RREQ.

```
Upon receiving RREQ with a fresh sequence number do {
    #improved route to sink
    if (RREQ_hop_count < current hop_count) {
        #to minimize collisions
        retransmit RREQ after a random delay
    #to support field vector reversal
    } elsif (RREQ_source is UPHILL_node) {
        #to minimize collisions
        retransmit RREQ after a random delay
        #notify neighbors about next_hop change
        } elsif (new next_hop != current next_hop &&
                RREQ_hop_count == current hop_count) {
                    send non-propagating RREQ message
                } else
        #no topology changes, drop message
        discard RREQ
}
```

1. Un mensaje inicial *Route Request* (RREQ) se envía desde el nodo dreno. Este mensaje se propaga completamente a través de la WSN, permitiendo a los diferentes nodos sensores generar vectores de rutas apuntando hacia el nodo dreno.

2. Al recibir los mensajes enviados por el dreno, cada nodo sensor almacena el valor del contador de saltos y el número de secuencia del último mensaje recibido. Desde el conjunto de vecinos que comparten el mismo (mínimo) contador de saltos hacia el dreno (nodos *downhill*), cada nodo toma uno de ellos como el siguiente salto (*next-hop*) para el reenvío de datos. Los nodos sensores también almacenan información acerca de otros vecinos (ambos nodos *peers* y *uphill*), basados en la información recolectada desde los mensajes escuchados. En resumen, la información almacenada por cada sensor es <*RREQ sequence number, hop count, next_hop, downhill nodes, peers, uphill nodes*>.

3. Debido a que el dreno puede ser móvil, este envía mensajes RREQ periódicamente para anunciar su presencia, a intervalos de tiempo regulares. Los intervalos pueden ser ajustados de acuerdo al grado de movilidad (un segundo por defecto). Estos mensajes permiten que los nodos sensores cercanos puedan detectar cualquier cambio en la posición del dreno, los cuales pueden iniciar una reconfiguración de la topología.

4. Los nodos sensores podrán anunciar al dreno mediante el reenvío de mensajes *broadcast*, en función de su contador de saltos y el número de secuencia de acuerdo al algoritmo 4.2.

A diferencia del protocolo DABR, la posibilidad de descartar mensajes RREQ en función de los parámetros comentados, reportará beneficios en las prestaciones del protocolo MRLG.

Cuadro 4.1: Campos de la tabla de encaminamiento.

Nombre	Descripción
rt_flags	Indicador de encaminamiento
rt_next_hop	Dirección destino
rt_num_seq	Número de secuencia
rt_hop_count	Número de saltos para alcanzar el destino
rt_last_hop_count	Último número de saltos
rt_prev_hop	Dirección fuente
rt_pkt_count	Contador de paquetes

4.3.3. Mantenimiento de la tabla de encaminamiento

Con la finalidad de evitar una actualización de ruta en todos los nodos, cada nodo mantiene una entrada en la tabla de encaminamiento, la cual se actualizará dependiendo de las condiciones evaluadas por el protocolo MRLG. Cada nodo almacena solo el primer salto hacia el dreno, y la distancia en saltos a la que se encuentra del dreno.

Un nodo que recibe un mensaje de actualización de ruta cambiará su entrada en la tabla si el número de secuencia (*num_seq*) es igual o más reciente, y el contador de saltos (*hop_count*) es menor que el almacenado. Cada entrada en la tabla de encaminamiento contiene los datos mostrados en la tabla 4.1.

Para el reenvío de paquetes, el nodo fuente elegirá de entre los nodos vecinos al que tenga el número de secuencia más reciente (excluyendo aquellos nodos de los cuales el nodo en cuestión es su *next_hop*), que tenga como *next_hop* la entrada con menor número de saltos. En caso de empate, se seleccionará aquella ruta que tenga menor contador de paquetes (*pkt_count*). Cada vez que se envía un paquete se incrementará el *pkt_count*.

4.3.4. Administración de enlaces

Cuando un nodo pierda el enlace, y tras realizar tres reintentos sin éxito, se borra ese nodo de la lista de vecinos. Posteriormente, de entre todos los nodos marcados con número de secuencia más reciente (excluyendo aquellos nodos de los cuales el nodo es su *next_hop*), se elije como posibles *next_hops* las entradas con menor número de saltos. A continuación se envía un mensaje *broadcast* notificando el *hop_count* y quiénes son los nuevos *next_hops*. Si no tiene *next_hop* válido, solo se inhibe de transmitir datos, esperando una actualización por parte de los nodos vecinos o directamente del dreno.

Debido a que el MRLG depende en gran medida de los mecanismos de restauración de rutas locales, los sensores con fallos podrían permanecer ocultos durante largos periodos de tiempo. Para evitar esta situación, el dreno podrá generar periódicamente un tipo especial de mensaje RREQ, que debe ser propagado a todos los nodos sensores en la WSN, eliminando así a estos tipo de sensores de la topología. Para maximizar el rendimiento y para mantener la sobrecarga de encaminamiento en valores muy bajos,

tales RREQs especiales deben tener un periodo entre mensajes mucho más grande que los mensajes predeterminados RREQ.

4.4. Sumario

En este capítulo se han presentado dos protocolos de encaminamiento propuestos: el DABR y el MRLG. Estos protocolos tienen como objetivo la minimización de la sobrecarga de encaminamiento en WSNs con drenos estáticos y móviles, respectivamente. El algoritmo DABR utiliza un sistema de descubrimiento de ruta basado en anuncio del dreno, que siempre se propaga por la WSN sin limitaciones. El dreno es el responsable de crear y mantener las rutas, y cualquier nodo sensor podrá enviar y reenviar paquetes hacia el dreno usando el procedimiento estándar: el nodo consulta su tabla de encaminamiento para ver si tiene una ruta valida hacia el dreno, y después envía la información usando dicha ruta. En caso de que no se tenga una ruta disponible, todo el tráfico se descarta hasta que la ruta sea restaurada.

El algoritmo MRLG también está basado en el envío de mensajes por parte del dreno pero, a diferencia del anterior, envía mensajes anunciando su presencia a los nodos que se encuentran dentro de su alcance, y a una tasa muy superior a la del protocolo DABR. Para evitar colapsar la WSN con tráfico de encaminamiento, los sensores cercanos al dreno comprueban si ha habido alguna variación en su número de saltos hacia el mismo, de tal manera que únicamente si se han detectado variaciones, se procede a actualizar la ruta mediante una propagación condicional de la nueva posición del dreno. De esta manera se logra dar soporte a cambios bruscos en la posición del dreno, evitando así al máximo introducir sobrecarga innecesaria en la red.

Capítulo 5

Evaluación de prestaciones

En este capítulo se presenta la evaluación de prestaciones de los protocolos DABR y MRLG propuestos utilizando la tecnología IEEE 802.15.4 bajo diferentes condiciones. Utilizamos varias métricas de simulación, como la tasa de pérdida de paquetes, el promedio del retardo extremo a extremo y la sobrecarga de encaminamiento. Determinamos el grado de efectividad del estándar IEEE 802.15.4 para el soporte de tareas de tiempo crítico en WSNs multi-salto, mostrando sus limitaciones en el tamaño y la cantidad de tráfico que fluye a través de la red. El uso de la herramienta generadora de eventos presentada previamente permite generar modelos de propagación de gas y fuego, que son usados como entrada en la herramienta de simulación ns-2 [72]. Esta herramienta también permite reconstruir los eventos usando las trazas de la simulación, para comparar las áreas afectadas reales y estimadas, y determinar la eficiencia del sistema propuesto.

5.1. Introducción

Las aplicaciones de WSNs han crecido significativamente en los años recientes, llegando a ser relevantes en muchas áreas de aplicación. La fiabilidad y baja latencia son algunas características que deben reunir las aplicaciones WSN demandadas. Particularmente, para ser capaces de actuar de acuerdo a los cambios observados en el medio ambiente tan pronto como sea posible, las aplicaciones WSN deberán detectar los eventos rápidamente y estar seguros de que la información recolectada por los nodos sensores es correcta, siendo el tiempo de respuesta un factor crítico en estas aplicaciones. En este capítulo nos centramos en aplicaciones WSN para la monitorización de ambientes en interiores y exteriores. La evaluación de prestaciones se realiza utilizando un sistema de monitorización de eventos en tiempo real, evaluando los tiempos de retardo en la comunicación. Para evaluar el desempeño de los protocolos DABR y MRLG se utiliza una herramienta modeladora de eventos que permite comparar la entrada y la salida de los eventos, y determinar el grado de precisión logrado en el proceso de monitorización.

5.2. Entorno de simulación basado en ns-2

El diseño y prueba a gran escala de las WSNs es una tarea bastante compleja, requiriendo del uso de un simulador de redes que permita modelar con precisión las diferentes capas de red involucradas. Actualmente hay varios simuladores de red disponibles que dan soporte a WSNs. Los más conocidos y representativos son NS-2 [72], J-SIM [77], Jist/Swans [78], NCTUns [79], Omnet++ [80], Ptolemy-II [81] y TOSSF [82], entre otros.

El simulador ns-2 cubre un gran número de aplicaciones, protocolos, tipos de redes, elementos de red y modelos de tráfico. Está basado en dos lenguajes: uno orientado a objetos (C++), y un intérprete OTcl (una extensión del Tcl orientada a objetos), utilizado para ejecutar *scripts* de comandos de usuario. Ns-2 tiene una gran biblioteca de objetos de red y protocolos. Estos tienen dos clases jerárquicas: la compilada en C++ y la interpretada en OTcl, con correspondencia uno a uno entre ellas. La jerarquía compilada de C++ permite lograr eficiencia en la simulación y rapidez en el tiempo de ejecución. Mediante *scripts* OTcl se pueden definir topologías de redes, y especificar protocolos y aplicaciones. Ns-2 es un simulador de eventos discreto, donde el avance del tiempo depende del número de eventos que son administrados por el planificador. Ns-2 produce resultados de los que se pueden obtener datos para todo tipo de mediciones sobre la simulación, o bien trazas específicas para visualizarlas con la herramienta *nam*, la cual produce una animación de la simulación.

Evaluar la efectividad de una WSN que soporta aplicaciones críticas a través de simulaciones requiere crear o adaptar protocolos de red para el simulador. Adicionalmente, para medir las prestaciones de los sistemas desde una perspectiva global, también requerimos herramientas de generación de eventos realistas que puedan ser usadas tanto como entrada del simulador, como de referencia a la hora de evaluar los resultados de salida. Por ejemplo, cuando evaluamos la efectividad de una WSN rastreando la posición de un intruso en tiempo-real, deberemos primero modelar el patrón de movilidad del intruso, y después usar este como entrada para el simulador.

5.3. Metodología

El procedimiento general seguido para la evaluación de las prestaciones de la arquitectura para redes de sensores propuesta se puede resumir de la siguiente forma. En primer lugar se describen los escenarios específicos para cada uno de los eventos a simular, incluyendo la movilidad del dreno (para el caso de evaluación del protocolo MRLG). Posteriormente utilizamos una tabla donde se registran todos los parámetros y rangos de valores a utilizar en la simulación, para configurar el fichero .tcl. Como resultado de la simulación se producen ficheros traza, que posteriormente son analizados con scripts en *Perl* y *awk*, y visualizados con la herramienta *nam* y *Gnuplot*. El fichero .tcl estará definido con todos los requisitos del usuario, incluyendo la topología de red, fuentes de tráfico, y tiempo para iniciar y detener la transmisión de paquetes a través del planificador de eventos. Los parámetros evaluados serán: la sobrecarga de encaminamiento, el retardo extremo-a-extremo y la tasa de pérdida.

Cuadro 5.1: Parámetros principales para la simulación de eventos WSNs.

Tipo de evento	Gas / Fuego
PHY/MAC	IEEE 802.15.4 / 2.4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	500 segundos
Área de simulación	200x200 metros (interior) / 1000x500 metros (exterior)
Topología	Grid
Protocolo de encaminamiento	DABR
Rango de transmisión	10 metros / 50 metros
Tamaño del paquete	50 bytes
Número de nodos	200

5.4. Evaluación del protocolo DABR en escenarios con dreno estático

Los escenarios evaluados tendrán desde 40 nodos hasta 400 nodos distribuidos en una topología en malla, donde cada uno de los escenarios contará con un solo dreno estático, que es quien recibirá la información enviada desde los distintos nodos fuente. Se evaluará el impacto de variar la tasa de inyección de paquetes y el número de nodos fuente en cada uno de los escenarios.

5.4.1. Carga de trabajo y escenarios de simulación

En los dos tipos de eventos analizados existe una propagación de gas o fuego a través del área monitorizada. El comportamiento difiere en que los sensores del área incendiada se consideran destruidos y son desactivados de la red de sensores tan pronto como son alcanzados por el fuego. Por esa razón, el número de nodos sensores activos en cada uno de los eventos se espera que sea diferente. Particularmente, en el caso de la propagación del gas, los nodos sensores activos se estarán incrementando de acuerdo a la expansión del gas. En el caso de la propagación del fuego, los nodos sensores detectan el humo producido por el fuego, y estos deben enviar la información lo antes posible, ya que los nodos sensores serán destruidos por el fuego, y por lo tanto dejarán de recibir y enviar información. Para los experimentos realizados, la configuración de los eventos simulados se define en la tabla 5.1.

Se utilizaron 200 nodos desplegados en un área de 1000 x 500 mts. Los rangos de velocidad de propagación del evento variaron desde 1 a 7 m/s. Las simulaciones fueron repetidas 175 veces, variando la velocidad de propagación del evento. Con respecto al tráfico, los nodos fuente se activan en tiempos específicos de acuerdo a las características del evento modelado, como se define en la herramienta generadora de eventos para WSNs.

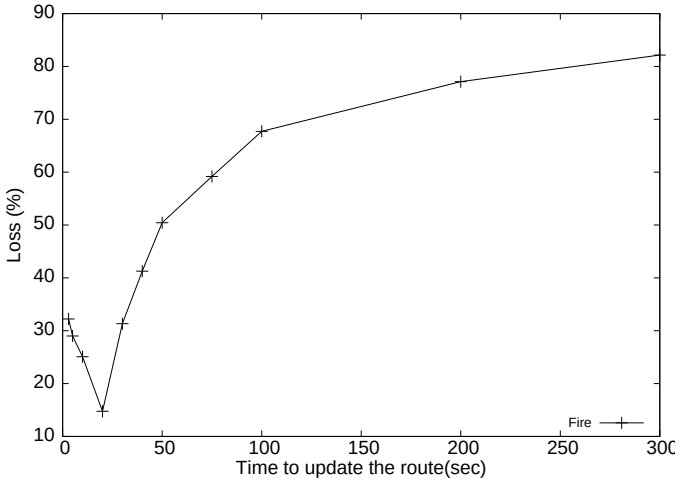


Figura 5.1: Tasa de pérdida de datos variando el intervalo de actualización cuando el fuego se afecta por una velocidad del viento de 3 m/s.

5.4.1.1. Tasa de actualización de rutas

El algoritmo de encaminamiento utilizado permite descubrir y mantener las rutas hacia el dreno, una vez que los nodos han sido desplegados y el nodo dreno se ha anunciado por medio de mensajes *broadcast*. En el caso de la propagación del fuego, los sensores son destruidos provocando interrupciones en la comunicación hacia los nodos vecinos y más alejados. Por esa razón, el dreno deberá periódicamente iniciar un nuevo proceso de descubrimiento de ruta, y así mitigar el efecto de escasez de sensores. Bajo estas condiciones, determinamos la tasa de actualización de encaminamiento óptima para garantizar una mayor tasa de entrega de paquetes, evitando demasiado tráfico de control que podría causar que la red se colapsara.

Para obtener el intervalo de actualización de ruta óptima, llevamos a cabo una serie de experimentos de simulación variando la tasa de actualización de anuncios del dreno. Los resultados se muestran en la figura 5.1, en la cual podemos observar claramente que, de acuerdo a los resultados obtenidos, el tiempo de actualización de ruta óptimo para la monitorización del fuego en WSNs es de 20 segundos para una velocidad del viento de 3 m/s. Aplicando el modelo de propagación de fuego explicado en el algoritmo 3.4 podemos encontrar el valor óptimo para el intervalo de actualización de encaminamiento para cualquier otra velocidad.

5.4.1.2. Medidas de latencia

En una aplicación WSN diseñada para eventos de tiempo crítico, la latencia es una característica muy importante y por lo tanto, la entrega de datos deberá satisfacer el valor de retardo máximo establecido para llegar hasta el nodo dreno. Llevamos a cabo un estudio del retardo asociado al tiempo en el que viajan los paquetes de datos

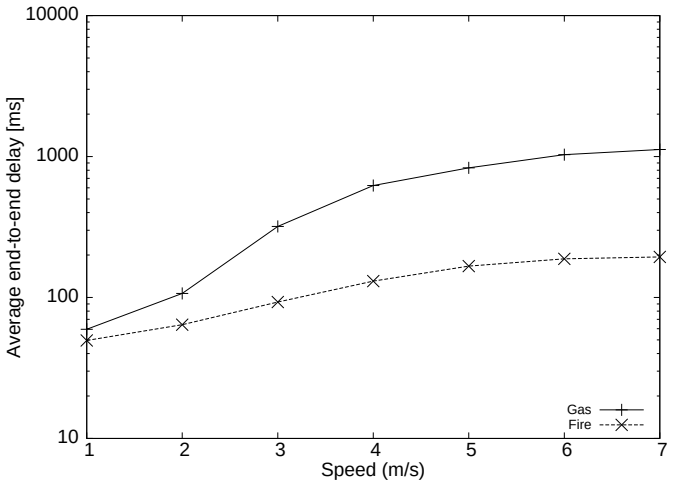


Figura 5.2: Retardo promedio extremo a extremo para eventos de gas y fuego.

que son inyectados desde la WSN, cuando la presencia de gas o fuego se detecta.

La velocidad del viento usada en la serie de simulaciones que fueron realizadas son basadas en [67], en el cual los estados que puede tener la velocidad del viento van desde 1 hasta 7 m/s. Con respecto a la propagación de gas en interiores, los valores de la velocidad de propagación del evento son normalmente más bajos que la velocidad del viento en exteriores. Sin embargo, para simplificar la comparación los experimentos se hicieron utilizando los mismos valores de las velocidades de propagación.

La figura 5.2 muestra el promedio de retardo extremo a extremo para los eventos de gas y fuego. El retardo promedio para la detección de gas va desde 60 ms a 1100 ms, conforme se incremente la velocidad. Tales diferencias se deben principalmente a un mayor número de nodos sensores que detectan el gas, congestionando el canal para el envío de paquetes al dreno. Esto provoca que los canales de comunicación lleguen a estar más saturados y se incremente la latencia. En esta misma figura podemos observar el comportamiento del retardo extremo a extremo para el fuego. Observe que comparado con el evento de gas, el promedio de retardo es más bajo. Esto es debido a que el número de sensores activos que inyectan tráfico es menor, ya que los sensores son consumidos por el fuego a medida que éste se desplaza.

5.4.1.3. Tasas de pérdida de paquetes

La cantidad de paquetes de datos que usamos para la evaluación y análisis de tiempo crítico fue de un paquete cada 12 segundos, por cada nodo fuente activo. Aunque esta cantidad genera poco tráfico, un gran número de sensores activos podría provocar que los canales del estándar IEEE 802.15.4 se lleguen a congestionar, debido a su baja capacidad (solo 250 kbit/s).

La figura 5.3 muestra el resultado en términos de tasa de paquetes para propa-

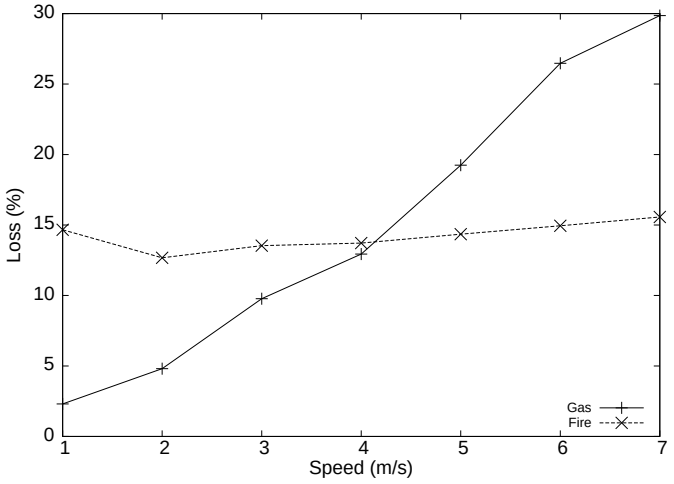


Figura 5.3: Porcentaje de pérdida de paquetes para eventos basados en gas y fuego.

gación de eventos de gas y fuego a diferentes velocidades. Para el modelo del gas, la pérdida tiene un crecimiento lineal desde 2 a 30 %. Este último valor del porcentaje, realizado con una velocidad de propagación de 7 m/s, logra que se incremente rápidamente la cantidad de nodos activos, haciendo que la expansión del gas sea detectada por casi todos los nodos en la WSN. A la velocidad de 7 m/s se llega a provocar la saturación en los canales de comunicación, lo cual provoca grandes pérdidas de paquetes. En el caso del fuego, la tasa de pérdida de paquetes se mantiene muy baja y más estable (cerca del 15 %) por dos razones: por un lado la velocidad es relativa al viento lo que significa que la velocidad real de propagación del fuego será mucho menor. Por otro lado, la destrucción de los nodos sensores causada por el fuego provoca que la cantidad de tráfico en la red se reduzca, lo cual ayuda a mitigar la pérdida.

Para entender mejor el comportamiento que se muestra en la figura 5.3, la figura 5.4 muestra el número de nodos activos involucrados en cada uno de estos eventos. Como podemos observar, los eventos de fuego generan un número más bajo de nodos sensores activos, debido a la continua destrucción de los mismos. Cuando estos eventos se ven afectados por la destrucción de sus nodos, se incrementa la pérdida pero, en general, el efecto a nivel global es mucho más limitado comparado con la situación del gas, donde el número de sensores activos llega a ser mucho mayor.

5.5. Evaluación de protocolos DABR y MRLG en escenarios con dreno dinámico

Para evaluar el rendimiento del protocolo MRLG bajo la tecnología IEEE 802.15.4, llevamos a cabo una serie de simulaciones utilizando el simulador de redes ns-2 [72]. La metodología seguida para conducir las pruebas fue la siguiente: se realizaron 4

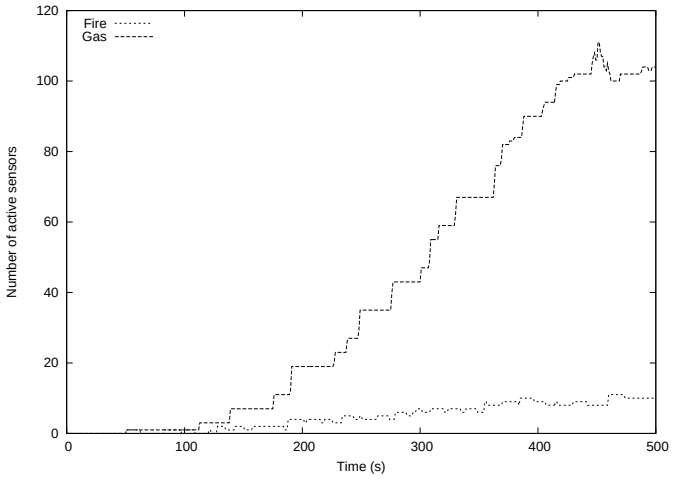


Figura 5.4: Número de nodos sensores activos en los eventos de gas y fuego con velocidad de propagación de 4 m/s.

conjuntos de pruebas, midiendo en cada uno de ellos el porcentaje de pérdida de datos, la latencia y la carga de encaminamiento. Cada uno de los escenarios de prueba considera que los nodos se despliegan en una topología de malla y que el dreno es móvil, localizado aleatoriamente en cualquier parte del escenario. El rango de transmisión utilizado para todos los nodos fue el máximo permitido por el estándar IEEE 802.15.4, que es de 10 metros. El modelo de propagación de radio fue *two-ray ground*. Otros parámetros relacionados con la configuración de los escenarios se muestran en las tablas de cada uno de los conjuntos de prueba descritos más adelante. Nuestras simulaciones están basadas en una serie de repeticiones, variando parámetros en cada uno de los conjuntos de prueba, con la finalidad de lograr una amplia evaluación del protocolo MRLG .

El protocolo MRLG es comparado con el protocolo de encaminamiento basado en anuncio del dreno el DABR.

5.5.1. En busca del mejor intervalo de descubrimiento de ruta para el DABR

Para determinar el mejor intervalo de descubrimiento de ruta, se realizaron una amplia serie de simulaciones variando la velocidad del dreno.

La figura 5.5 muestra los resultados obtenidos, variando la velocidad de desplazamiento del dreno desde 1 a 10 m/s. Como podemos observar, la pérdida de paquetes para 1 y 3 segundos es muy alta. Sin embargo, se utilizó un intervalo de anuncios por parte del dreno de 5 segundos, ya que es el valor más cercano al intervalo de envío de mensajes que se utilizó en el protocolo MRLG.

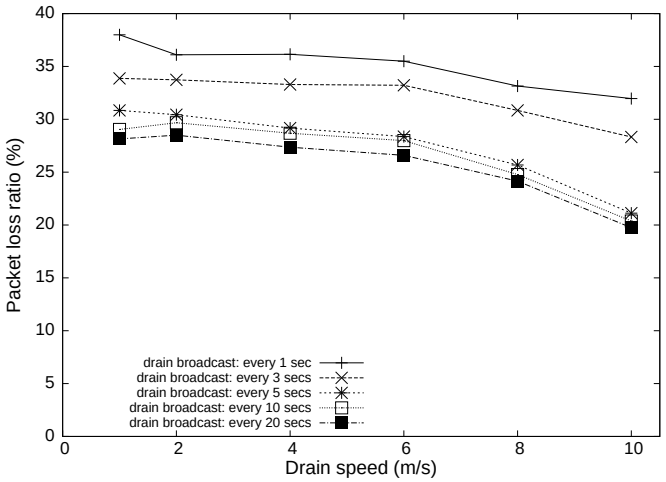


Figura 5.5: Tasa de pérdida obtenida con el protocolo de encaminamiento basado en anuncio del dreno, variando la velocidad de desplazamiento del dreno.

5.5.2. Impacto del número de nodos fuente

En este primer conjunto de pruebas analizamos el comportamiento de las WSNs cuando se incrementa el número de fuentes de tráfico.

La tabla 5.2 presenta los parámetros más representativos usados en las simulaciones, donde el propósito es medir el rendimiento del protocolo MRLG al variar el número de nodos que inyectan tráfico en la red. El número de nodos sensores que inyectan tráfico se incrementa desde 1 hasta 40 nodos, mientras que el número total de nodos se mantiene fijo en 200. El espacio físico en el que están distribuidos los nodos sensores es de 140x140 metros, y el tiempo de duración es de 600 segundos para cada simulación.

La figura 5.6 muestra la tasa de pérdida de paquetes a medida que aumenta el número de fuentes. La tasa de pérdida para el protocolo MRLG, se incrementa desde el 5 % hasta el 25 %, mientras que para el caso del protocolo DABR, la tasa de pérdida se incrementa drásticamente hasta cerca del 70 % cuando la cantidad de nodos fuente llega a ser 40.

La figura 5.7 muestra el retardo promedio extremo a extremo, para ambos protocolos de encaminamiento. Podemos observar que el promedio de retardo es menor en el protocolo MRLG, siendo aproximadamente un 10 % superior el retardo promedio que para el protocolo DABR.

En la figura 5.8 (a), podemos observar la sobrecarga de encaminamiento absoluto para ambos protocolos. Mientras que los nodos fuente generan la misma cantidad de paquetes en ambos protocolos, la sobrecarga de encaminamiento se mantiene estable en diez mil paquetes para el MRLG. Para el protocolo DABR se incrementa desde treinta y dos mil hasta cuarenta y tres mil paquetes para el rango de nodos fuente de

Cuadro 5.2: Parámetros de simulación variando la cantidad de nodos fuente.

Número de nodos	200
PHY/MAC	IEEE 802.15.4 / 2,4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	600 segundos
Área de simulación	140x140 metros
Topología	Grid
Protocolo de encaminamiento	MRLG/DABR
Rango de transmisión	10 metros
Tamaño del paquete	50 bytes
Número de fuentes de tráfico	1, 3, 7, 10, 15, 20, 25, 30, 35 y 40
Carga de tráfico	0,2 pqt/s

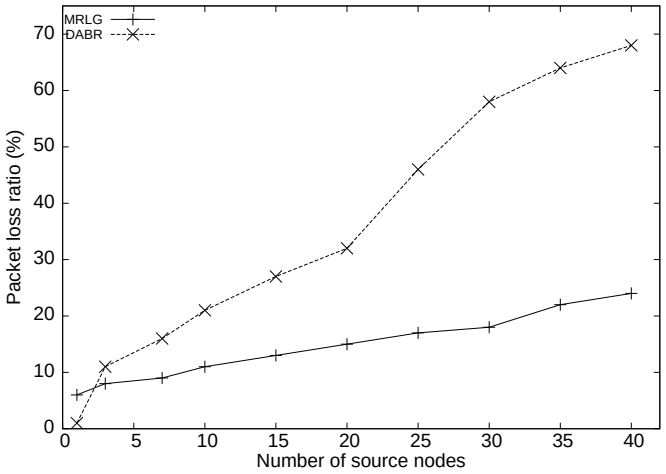


Figura 5.6: Tasa de pérdida variando la cantidad de nodos fuente.

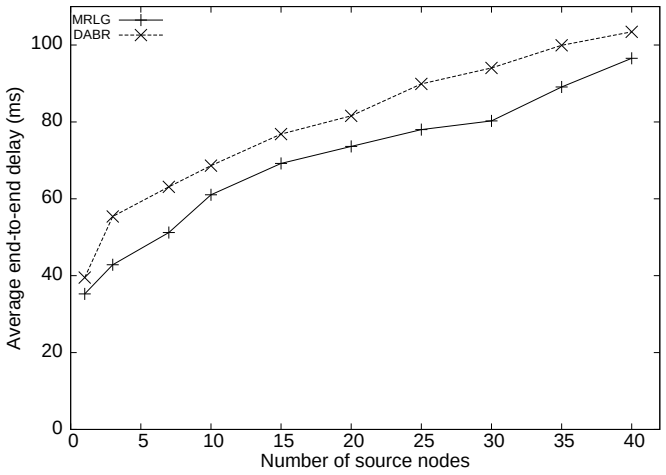


Figura 5.7: Retardo promedio variando la cantidad de nodos fuente.

1 a 40, respectivamente. La figura 5.8 (b), muestra la sobrecarga de encaminamiento normalizado para ambos protocolos, donde la carga del protocolo DABR es superior, en comparación con la carga de encaminamiento del protocolo MRLG.

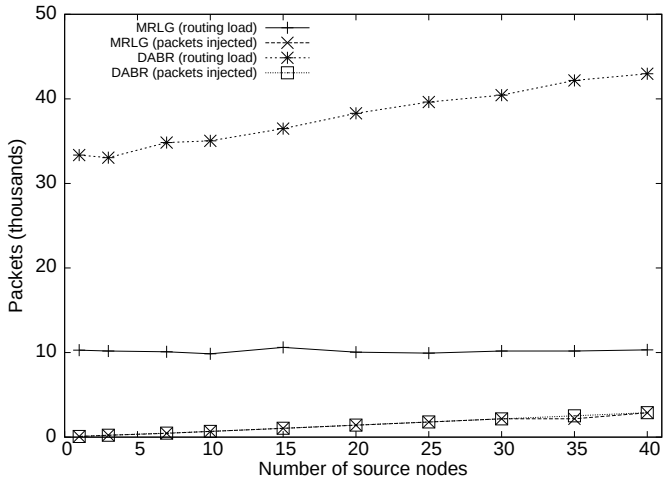
5.5.3. Evaluando el impacto del tráfico en la red

En este segundo conjunto de pruebas, analizamos el comportamiento de las WSNs cuando se incrementa la cantidad de tráfico inyectado por los nodos fuente. Los parámetros utilizados en estas simulaciones se muestran en la tabla 5.3. El número de nodos sensores que inyectan tráfico (nodos fuente) es fijo, e incrementamos la tasa de inyección de paquetes por nodo fuente. El número total de nodos también se mantiene fijo en 200.

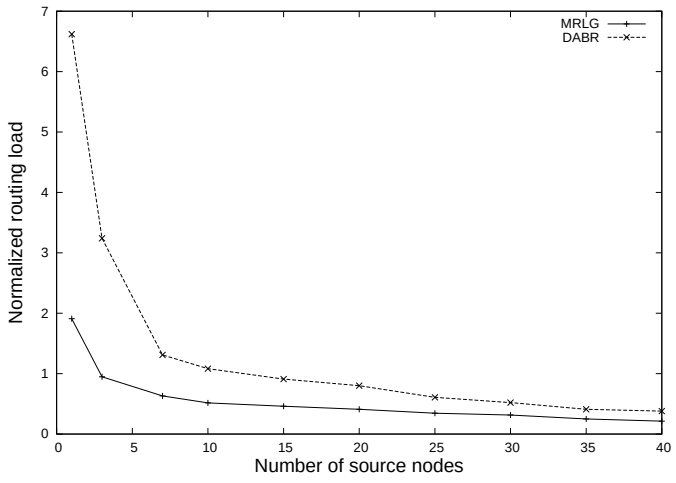
La figura 5.9 muestra el rendimiento obtenido con ambos protocolos respecto a la tasa de pérdida. En la gráfica podemos observar que el porcentaje de pérdida para el protocolo DABR es 300 % superior respecto a la pérdida que se experimenta cuando adoptamos el protocolo MRLG.

La figura 5.10 muestra el resultado obtenido del retardo promedio extremo a extremo, para ambos protocolos de encaminamiento, cuando varía la carga. Podemos observar que el retardo promedio es menor en el protocolo MRLG, y que incluso a partir de la inyección de 0.2 paquetes/segundo, el retardo es cada vez mayor en el protocolo DABR, llegando a 350 ms para el caso de inyección de un paquete por segundo.

En la figura 5.11 (a) podemos observar el número de paquetes de encaminamiento inyectados. En ambos protocolos los paquetes generados son los mismos, como nos muestra la gráfica, siendo muy alta la sobrecarga de encaminamiento del protocolo DABR con respecto al protocolo MRLG.



a)



b)

Figura 5.8: Sobrecarga de encaminamiento variando el número de nodos fuente: a) número de paquetes de encaminamiento inyectados y b) carga de encaminamiento normalizada.

Cuadro 5.3: Parámetros de simulación para evaluar el impacto del tráfico en la red.

Número de nodos	200
PHY/MAC	IEEE 802.15.4 / 2,4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	600 segundos
Área de simulación	140x140 metros
Topología	Grid
Protocolo de encaminamiento	MRLG/DABR
Rango de transmisión	10 metros
Tamaño del paquete	50 bytes
Número de fuentes de tráfico	20
Carga de tráfico	0,049; 0,1; 0,142; 0,2; 0,5 y 1 paq/s

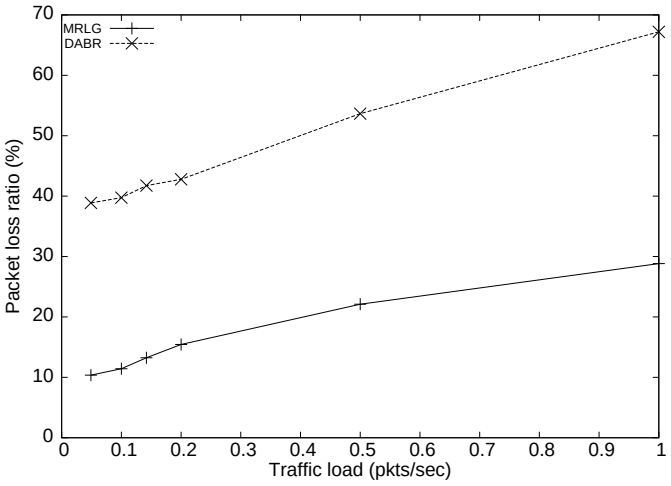


Figura 5.9: Tasa de pérdida variando la carga.

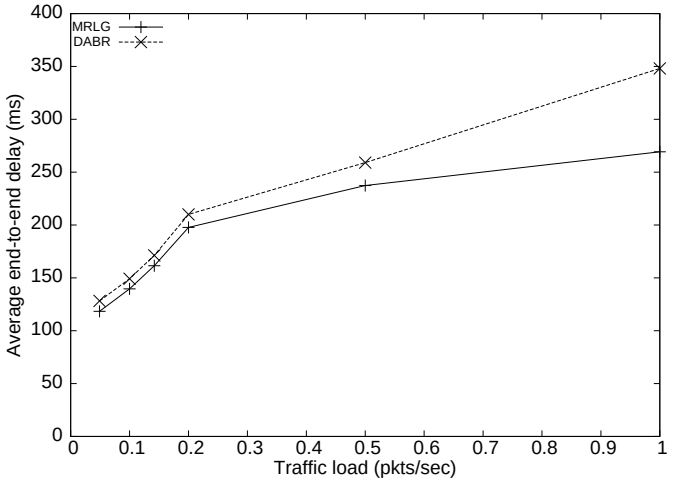


Figura 5.10: Retardo promedio variando la carga.

En la figura 5.11 (b) se muestra la sobrecarga de encaminamiento normalizado, mostrándonos nuevamente esta gráfica que, para esta serie de simulaciones, la sobrecarga de encaminamiento es superior para el protocolo DABR.

En general, los resultados obtenidos muestran claramente las ventajas del protocolo MRLG respecto a DABR.

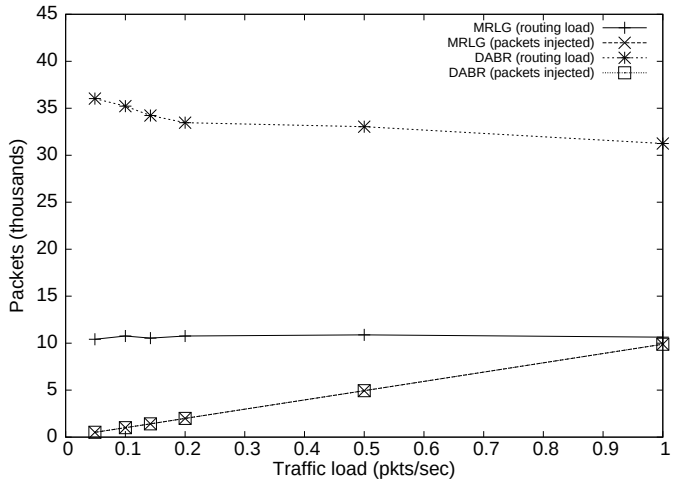
5.5.4. Análisis de escalabilidad

Con el conjunto de simulaciones realizadas en esta sección queremos evaluar y analizar la escalabilidad de las WSNs bajo el protocolo MRLG, variando el área de simulación y manteniendo fija la cantidad de nodos fuente, así como la cantidad de paquetes inyectados por nodo. La tabla 5.4 muestra los parámetros usados para esta serie de simulaciones.

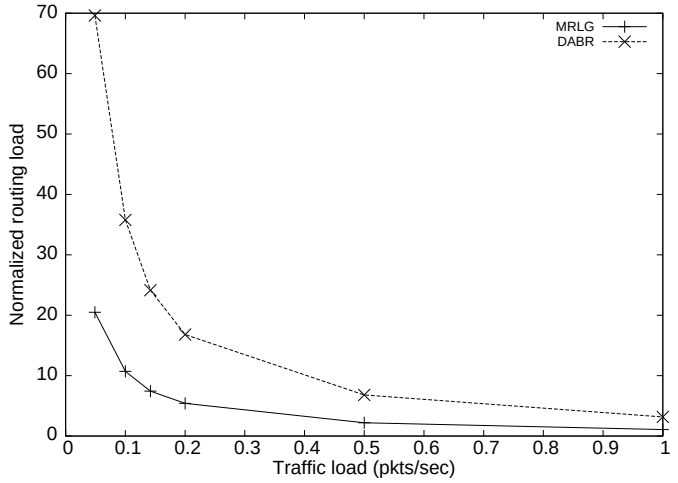
La figura 5.12 muestra la tasa de pérdida de paquetes para los diferentes escenarios analizados. El protocolo DABR tiene una mayor pérdida que el protocolo MRLG, ya que el porcentaje de pérdida para el protocolo MRLG va desde un 6 % hasta un 15 % aproximadamente, para los escenarios evaluados desde 80 hasta 400 nodos, mientras que la pérdida inicial con el protocolo DABR, para este mismo conjunto de escenarios, va desde 22 % hasta 55 %, aproximadamente.

La figura 5.13 muestra el resultado obtenido respecto al retardo promedio extremo a extremo, para ambos protocolos de encaminamiento y para los diferentes escenarios analizados. El retardo promedio obtenido con DABR presenta valores muy por encima del protocolo MRLG, obteniendo un retardo promedio en este último de 50 a 150 milisegundos para el rango de escenarios de 80 a 400 nodos.

En la figura 5.14 (a) podemos observar el número de paquetes encaminamiento además de los paquetes de datos inyectados en la red. En ambos protocolos de enca-



a)



b)

Figura 5.11: Sobrecarga de encaminamiento variando la tasa de inyección de paquetes por nodo fuente: a) número de paquetes de encaminamiento inyectados y b) carga de encaminamiento normalizada.

Cuadro 5.4: Parámetros de simulación para evaluar la escalabilidad del protocolo MRLG.

Número de nodos	80, 120, 160, 200, 300 y 400
PHY/MAC	IEEE 802.15.4 / 2.4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	600 segundos
Área de simulación	91x91, 112x112, 126x126, 140x140, 175x175 y 210x210 metros
Topología	Grid
Protocolo de encaminamiento	MRLG/DABR
Rango de transmisión	10 metros
Tamaño del paquete	50 bytes
Número de fuentes de tráfico	20
Carga de tráfico	0,2 paq/s

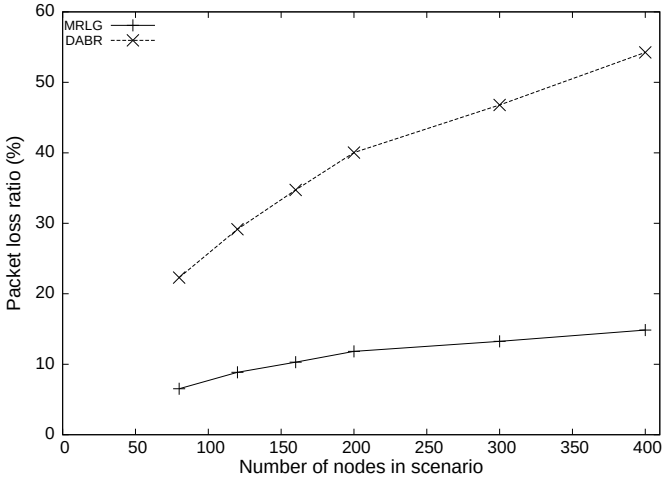


Figura 5.12: Tasa de pérdida variando el número de nodos por escenario.

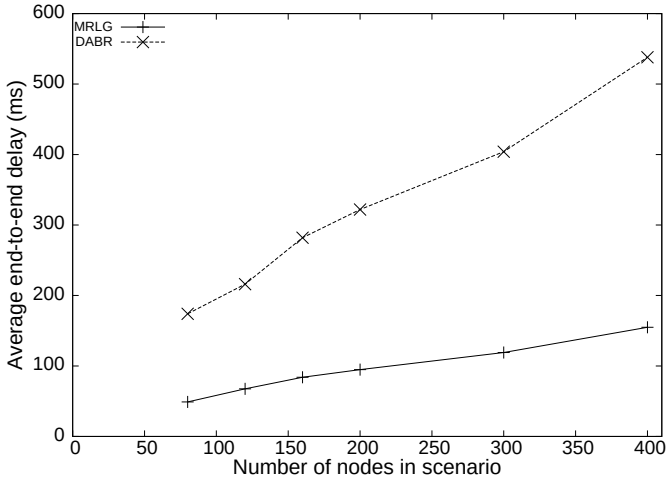


Figura 5.13: Retardo promedio variando el número de nodos por escenario.

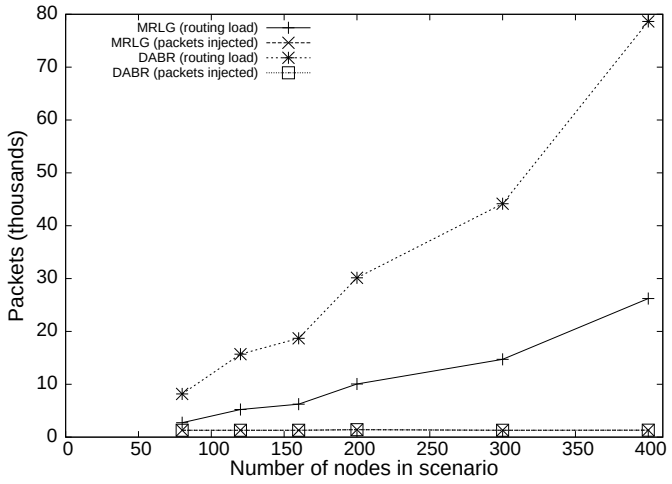
minamiento los paquetes generados son los mismos, como nos muestra la gráfica, y nuevamente la carga de encaminamiento del protocolo DABR es más alta, en comparación con el protocolo MRLG. Los valores de sobrecarga de encaminamiento para el protocolo MRLG crecen desde 2723 hasta 26214, mientras que con el protocolo DABR los valores crecen desde 8169 hasta 78642. En la figura 5.14 (b) se muestra la carga de encaminamiento normalizado, quedando evidente también en esta gráfica que la carga de encaminamiento es superior para el protocolo de encaminamiento basado en anuncio del dreno (DABR).

5.5.5. Capacidad de adaptación con distintas velocidades del dreno

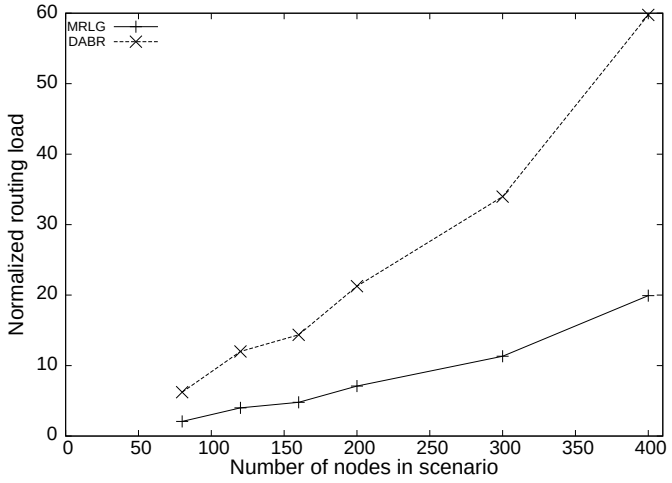
En esta sección vamos a mostrar los resultados de rendimiento para los protocolos DABR y MRLG cuando la velocidad del dreno está dentro del rango de 1 a 10 m/s. Los parámetros utilizados para este conjunto de simulaciones se muestra en la tabla 5.5.

La tasa de pérdida para las simulaciones variando la velocidad del dreno se presentan en la figura 5.15. La tasa de pérdida para el protocolo MRLG está muy por debajo en comparación con el protocolo DABR, teniendo un porcentaje de pérdida inferior al 15 %. El valor de pérdida se reduce un poco más en ambos protocolos cuando el dreno tiene una velocidad de 10 m/s, debido a que el dreno requiere hacer menos actualizaciones al pasar mas rápido cerca de los nodos, evitando que algunos nodos emitan mensajes *broadcast* para actualizar sus rutas.

La figura 5.16 muestra el resultado obtenido para el retardo promedio extremo a extremo y para ambos protocolos de encaminamiento, para las diferentes velocidades del dreno analizadas en esta sección. El porcentaje de retardo promedio para el



a)



b)

Figura 5.14: Sobrecarga de encaminamiento variando el número de nodos por escenario: a) número de paquetes de encaminamiento inyectados y b) carga de encaminamiento normalizada.

Cuadro 5.5: Parámetros de simulación para evaluar el protocolo MRLG variando la velocidad del dreno.

Número de nodos	200
PHY/MAC	IEEE 802.15.4 / 2,4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	600 segundos
Área de simulación	140x140 metros
Topología	Grid
Protocolo de encaminamiento	MRLG/DABR
Rango de transmisión	10 metros
Tamaño del paquete	50 bytes
Número de fuentes de tráfico	20
Carga de tráfico	0,2 pkt/s
Velocidad del dreno	1, 2, 4, 6, 8 y 10 m/s

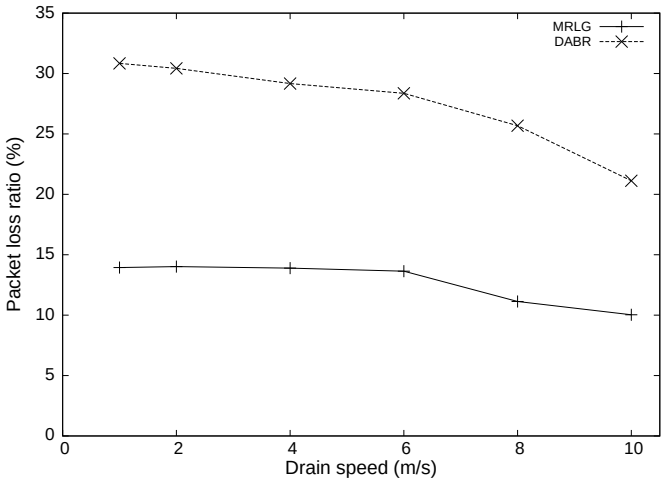


Figura 5.15: Tasa de pérdida variando la velocidad del dreno.

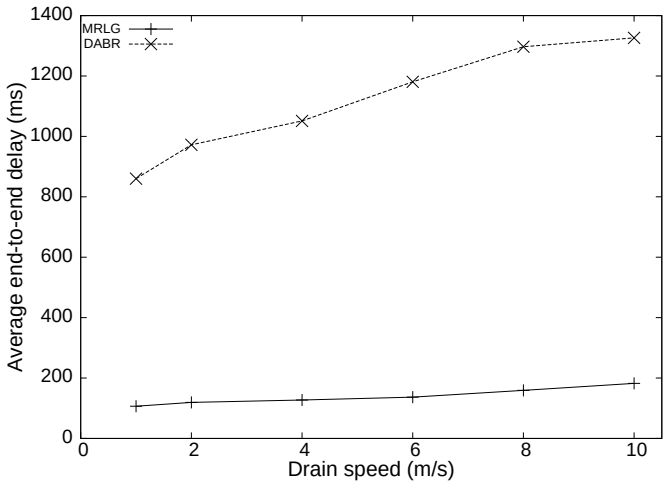


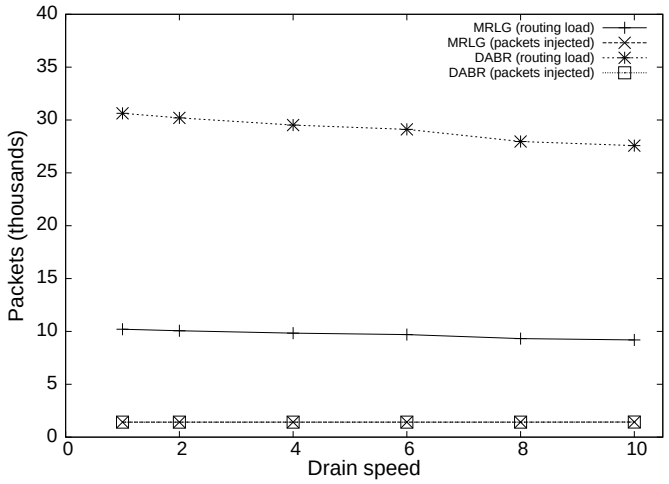
Figura 5.16: Retardo promedio variando la velocidad del dreno.

protocolo DABR alcanza valores desde 800 ms hasta 1300 ms, mientras que, para el protocolo MRLG, los valores del retardo promedio son inferiores a 200 ms. Obviamente, estos valores se acercan bastante más a los requisitos típicos de tiempo real que los valores obtenidos con DABR.

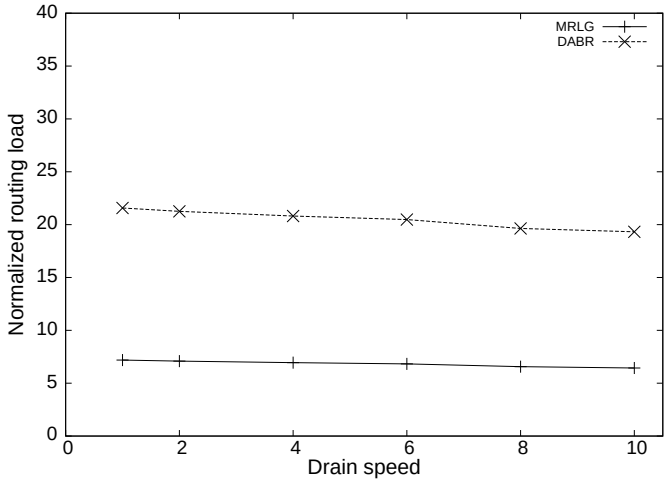
En la figura 5.17 (a) podemos observar el número de paquetes encaminamiento/inyectados. Nuevamente el comportamiento de ambos protocolos de encaminamiento, mantienen la misma relación que las pruebas anteriores, donde los paquetes generados son los mismos, y nuevamente la sobrecarga de encaminamiento es más alta para el protocolo DABR. En la figura 5.17 (b) se muestra la sobrecarga de encaminamiento normalizado, siendo evidente que la sobrecarga de encaminamiento es superior para el protocolo DABR.

5.6. Medidas de precisión de los eventos generados con la herramienta modeladora

En esta sección analizamos la estimación del error como una función del tiempo, al intentar reconstruir los eventos de propagación de gas y fuego, así como la reconstrucción de la trayectoria de intrusos. El cálculo de estimación de la precisión del sistema propuesto se lleva a cabo con un conjunto de resultados de pruebas, realizadas mediante una serie de simulaciones utilizando el simulador ns-2. Los datos recibidos de los diferentes nodos fuente son tomados en cuenta para la reconstrucción del evento, considerando la posición de los nodos activos en el escenario.



a)



b)

Figura 5.17: Sobrecarga de encaminamiento variando la velocidad del dreno: a) número de paquetes de encaminamiento inyectados y b) carga de encaminamiento normalizada.

5.6.1. Escenarios de propagación de gas y fuego

La estimación de error asociado con los dos tipos de eventos en función del tiempo se muestra en las figuras 5.18 y 5.19. En la figura 5.18 se muestran las gráficas para estimar el límite del error ante eventos de propagación de gas y fuego, respectivamente. La figura 5.19 muestran el área de error estimado para los eventos de gas y fuego, respectivamente, tal y como se ha definido en la sección anterior.

Para el caso de propagación de gas, encontramos en la figura 5.18 (a) que el error es de 78 m a una velocidad de 3 m/s, y 52 metros a una velocidad de 5 m/s, decreciendo gradualmente con el tiempo. En el caso de la propagación del fuego, podemos observar en la figura 5.18 (b) que el error para la propagación del fuego a velocidades de 3 y 5 m/s, es de cerca de 8 metros para ambas velocidades en el comienzo de la simulación, alcanzando 28 y 32 metros, respectivamente, en el instante de tiempo 500 segundos. Observemos que la propagación del fuego difiere de la propagación del gas debido a la destrucción de los nodos sensores, significando que los enlaces en la WSN se interrumpen, llegando a ser necesaria una actualización de ruta, lo cuál provoca pérdida de datos al no tener rutas válidas para el envío de la información, lo que incrementa la inexactitud en el proceso de rastreo.

La figura 5.19 en las gráficas (a) y (b) muestran el error del área para los eventos de gas y fuego a diferentes velocidades. En la propagación de gas con velocidad de 3 y 5 metros por segundo, podemos apreciar un comportamiento de error similar, que es de 27 y 24 %, respectivamente, decrementando ambas hasta acercarse al 10 % en un tiempo de 380 segundos. Este es el instante de tiempo (380 segundos) a partir del cual nuevamente el error de estimación va incrementándose debido a que los eventos han alcanzado el número máximo de nodos sensores involucrados en la simulación. En el caso del error de estimación del área de propagación del fuego, la tasa de error varía del 5 % hasta el 24 % y 29 % con velocidades del viento de 3 y 5 m/s, respectivamente, incrementándose progresivamente.

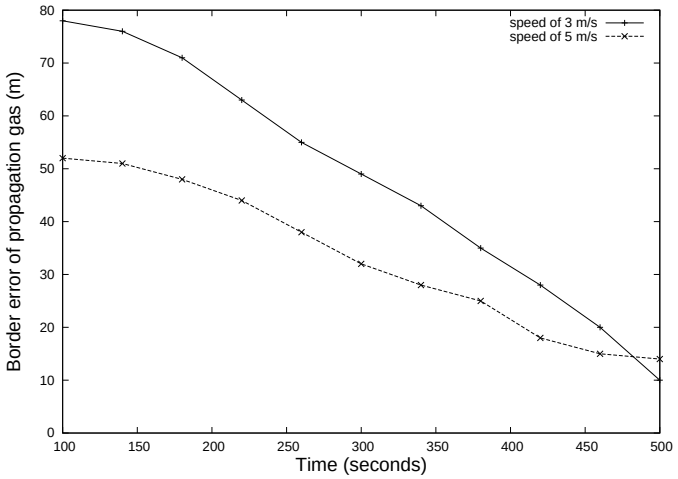
5.6.2. Escenarios de seguimiento de intrusos

Para evaluar la precisión del algoritmo de seguimiento de intrusos propuesto, hemos llevado a cabo series de simulaciones donde variamos los parámetros más críticos, tales como la velocidad del intruso, la trayectoria del intruso, la velocidad del dreno, el tamaño de la red y la frecuencia de generación de mensajes. También calculamos el error de seguimiento mediante la medición de la distancia euclídea entre la posición estimada y la posición real del intruso.

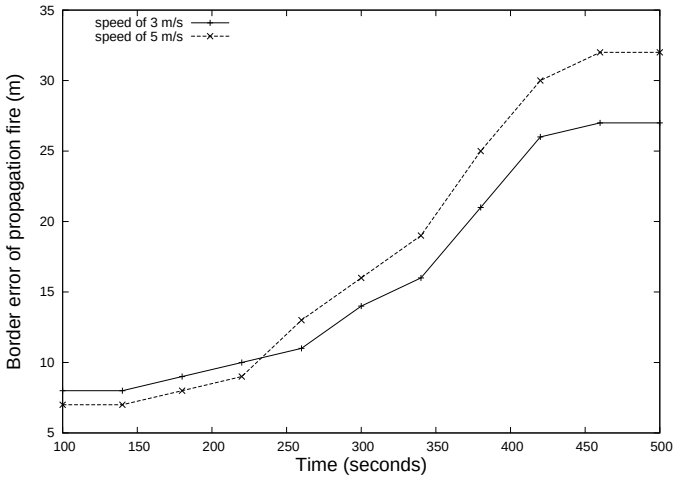
El procedimiento de evaluación es el siguiente: empezamos por la evaluación del impacto del protocolo de encaminamiento seleccionado. Después, evaluamos el impacto de los patrones de movilidad de un intruso con respecto a la precisión de seguimiento. Finalmente, analizamos el impacto de la movilidad del dreno.

5.6.2.1. Configuración de la Simulación

Para esta serie de experimentos utilizamos el simulador de redes ns-2. En cada una de las pruebas hay un único dreno móvil que, utilizando la secuencia de posiciones

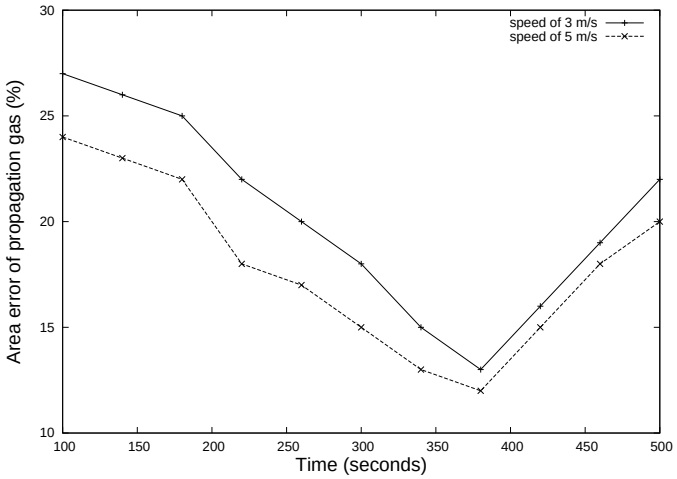


a)

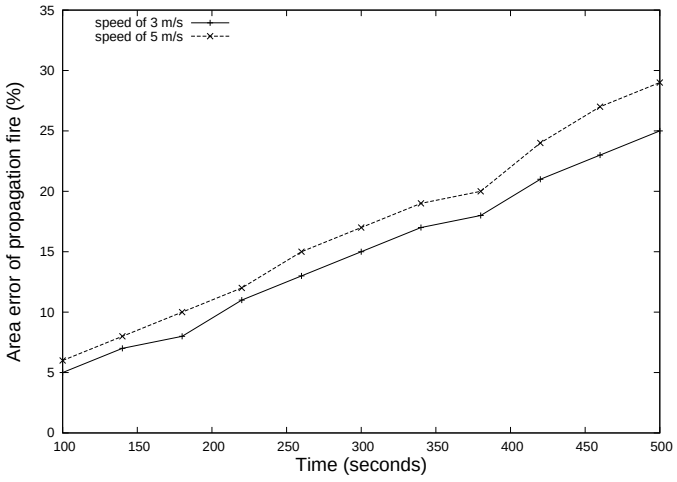


b)

Figura 5.18: Comportamiento del error estimado en el borde.



a)



b)

Figura 5.19: Comportamiento del error estimado en el área.

Cuadro 5.6: Parámetros de referencia para las simulaciones.

Número de nodos	200
PHY/MAC	IEEE 802.15.4 / 2,4 GHz
Tipo de tráfico	CBR
Tiempo de simulación	500 segundos
Área de simulación	200x100 metros
Topología	Grid
Protocolo de encaminamiento	MRLG / DABR
Rango de transmisión	10 metros
Tamaño del paquete	50 bytes
Carga de tráfico	0,2 pqts/s
Velocidad del intruso	4 m/s
Velocidad del dreno	4 m/s

estimadas que el sistema ofrece respecto al intruso, constantemente se mueve hacia el mismo como si intentara atraparlo en una búsqueda real.

Los parámetros de simulación usados por defecto en los diferentes experimentos son los que se muestran en la tabla 5.6, a menos que se indique lo contrario. Desplegamos 200 nodos siguiendo una topología de malla regular, y la distancia entre sensores se ha establecido en 10 metros [15] ya que las comunicaciones de radio se basan en el estándar IEEE 802.15.4. El modelo de propagación de radio adoptado es el *two-ray ground*. Respecto al dreno, está localizado inicialmente en la parte superior izquierda del escenario, y en base a la estimación de la posición hecha relativa al intruso, se mueve hacia él. Los protocolos de encaminamiento usados para las pruebas son DABR [83] y MRLG [76].

La metodología utilizada para realizar las pruebas fue la siguiente: primero generamos el patrón de movilidad del intruso a lo largo del área monitorizada. Después calculamos el instante de tiempo en el cual los diferentes sensores son activados por el acercamiento del intruso. Adoptamos un sistema de detección basado en sensores binarios, en donde cada sensor enviará un mensaje de intruso detectado al dreno inmediatamente después de la detección. Si el intruso se mantiene dentro de la zona de detección de los sensores, cada sensor activo continuará informando de la presencia del intruso cada 5 segundos por defecto. A partir de los diferentes informes reunidos relativos al intruso, el dreno estima periódicamente la posición actual del intruso, y de forma dinámica se dirigirá hacia su objetivo.

5.6.2.2. Impacto del protocolo de encaminamiento

En esta primera serie de pruebas realizamos varias simulaciones, para determinar el impacto del protocolo de encaminamiento utilizado en la estimación del error de la posición de intrusos. Cabe recordar que, mientras que la primera estrategia de encaminamiento (DABR) genera mensajes periódicos de actualización que se propagan

Cuadro 5.7: Parámetros de simulación al medir el impacto del protocolo de encaminamiento elegido.

Número de nodos	40, 80, 120, 160, 200, 300 y 400
Protocolo de encaminamiento	DABR/MRLG
Carga de tráfico	0,05; 0,066; 0,083; 0,011; 0,166; 0,333; 0,5 y 1 paq/s
Ruta del intruso	Aleatoria

a lo largo de toda la WSN, la segunda de ellas (MRLG) restringe la propagación de mensajes tanto como sea posible, quedándose generalmente bloqueado por los vecinos del dreno. Estos dos protocolos, tan heterogéneos entre sí, presentan diferentes grados de eficacia y costes generales de encaminamiento, proporcionando información importante sobre el impacto del encaminamiento en términos de rendimiento. Los parámetros de simulación adoptados son los que se muestran en la tabla 5.6. Sin embargo, variamos algunos de estos parámetros para obtener una mayor comprensión en cuanto a la dependencia de cada uno de ellos en términos de rendimiento. La tabla 5.7 muestra estos parámetros de simulación que pueden variar, dependiendo del experimento, así como los valores adoptados.

La figura 5.20 muestra el error de estimación medio obtenido al variar diferentes parámetros de simulación. La figura 5.20 (a) muestra que el impacto en términos de error de estimación es mínima. Además, nos encontramos con que MRLG mejora la precisión en cuanto a la estimación de posición, introduciendo un error de aproximadamente 5 metros, mientras que para el protocolo DABR el error estimado es de unos 7 metros. La figura 5.20 (b) muestra el error experimentado al variar la velocidad del intruso de 1 a 7 m/s. Nos encontramos que, para el protocolo MRLG, el error se incrementa desde algo menos que 2 hasta casi 6 metros, mientras que para el protocolo DABR el error de estimación aumenta de 2 a 10 metros. Esto ocurre debido a que la diferencia entre la posición actual del intruso y la estimada llega a ser mayor a medida que aumenta la velocidad del intruso. En cuanto a la escalabilidad del tamaño de la red, la figura 5.20 (c) muestra el impacto de aumentar el número de nodos en el escenario desde 40 hasta 400. La media de la estimación de error se incrementa de 1.5 a 8 metros cuando se utiliza el protocolo MRLG, mientras que para el protocolo DABR el error medio estimado aumenta de 2 a 10 metros. Esto significa que la latencia adicional asociada con el mayor numero de saltos necesarios para alcanzar el dreno aumenta el error estimado para la posición del intruso, como se esperaba.

La figura 5.21 muestra los resultados obtenidos en cuanto a sobrecarga de encaminamiento para el mismo conjunto de experimentos. La figura 5.21 (a) muestra que un aumento de la cantidad de tráfico inyectado tiene un impacto mínimo en la sobrecarga de encaminamiento, especialmente para el protocolo MRLG. Esto es de esperar ya que la topología de red se mantiene de forma proactiva, independientemente de la cantidad de tráfico de datos que realmente se transmite. La figura 5.21 (b) muestra

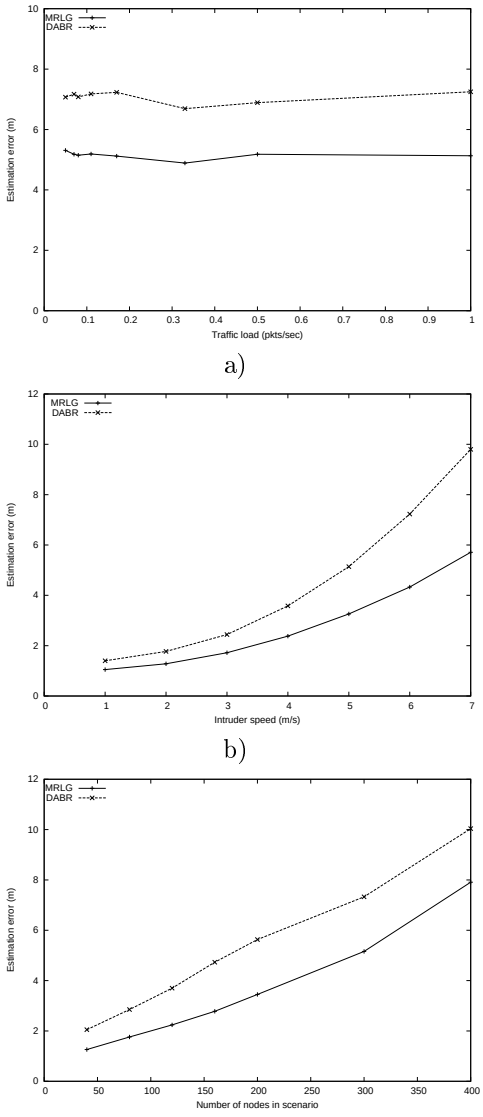


Figura 5.20: Promedio de error estimado utilizando los protocolos DABR y MRLG cuando se varía: a) cantidad de tráfico inyectado, b) velocidad del intruso, y c) número de nodos en el escenario.

Cuadro 5.8: Parámetros de simulación cuando varia los patrones de movilidad.

Número de nodos	200
Protocolo de encaminamiento	MRLG
Carga de tráfico	0,2 paq/s
Velocidad del intruso	1, 2, 3, 4, 5, 6 y 7 m/s
Patrón de movilidad del intruso	Recta, aleatoria y curva

la sobrecarga de encaminamiento al variar la velocidad del intruso. Encontramos que este parámetro tiene poco impacto en los costes generales de encaminamiento, aunque se observan diferencias significativas entre los protocolos MRLG y DABR, siendo que este último presenta unas tres veces la sobrecarga introducida por el primero. La figura 5.21 (c) muestra la sobrecarga de encaminamiento al variar la cantidad de nodos sensores en el escenario, para la misma densidad. El objetivo es observar las propiedades de los protocolos en términos de escalabilidad. Los resultados evidencian que el MRLG es mucho más escalable que el DABR ya que introduce un aumento lineal del tráfico de encaminamiento, mientras que para el DABR el incremento se acentúa cuando el número de nodos en los escenarios crece hasta 400.

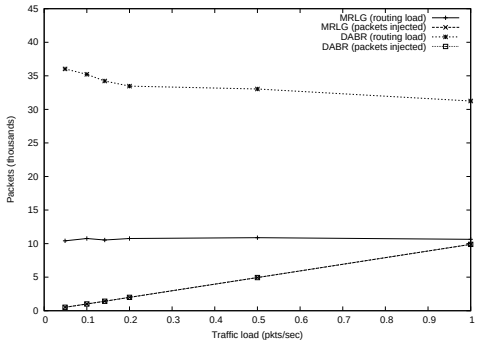
Los resultados presentados anteriormente hacen hincapié en las ventajas de utilizar un protocolo de encaminamiento que presente una baja sobrecarga de encaminamiento. Se detectó que la reducción de la sobrecarga general de encaminamiento permite reducir el error de precisión de seguimiento de intrusos debido a que la ocupación del canal se hace más baja. Se ha detectado también un impacto positivo del protocolo MRLG, el cual permite reducir tanto la pérdida de paquetes como el retardo extremo-a-extremo, lo que explica las mejoras logradas. En las secciones que siguen se realizan pruebas únicamente con el protocolo MRLG, debido a su mejor comportamiento.

5.6.2.3. Impacto de los patrones de movilidad del intruso

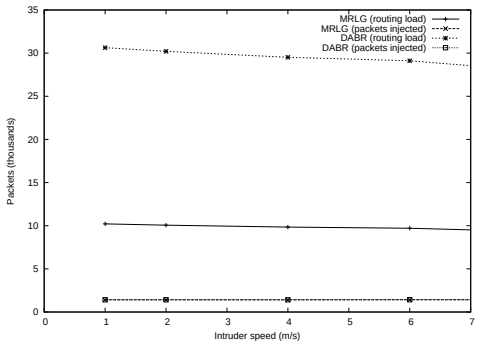
En este segundo conjunto de experimentos nos centraremos en el impacto del patrón de movilidad de un intruso y en la velocidad de precisión de rastreo. La tabla 5.8 resume los diferentes parámetros usado en las simulaciones presentadas a continuación, respecto a las de la sección anterior.

La figura 5.22 muestra los resultados obtenidos en términos de rendimiento. En general, verificamos que el error de estimación se incrementa conforme la velocidad del intruso varía de 1 a 7 m/s. Este valor era de esperar debido al tiempo requerido por los informes de detección para viajar hasta al dreno, así como por los retardos introducidos por el algoritmo de agregación de datos que está presente en el dreno, lo que le hacen propenso a hacer estimaciones de posición menos confiables a altas velocidades.

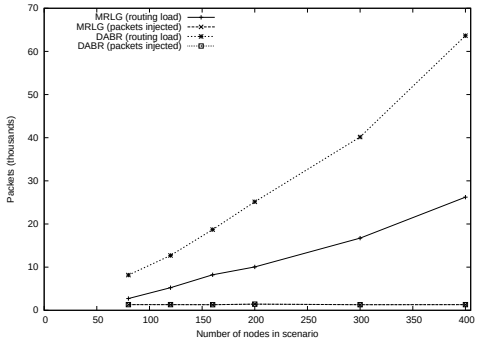
Comparando los diferentes patrones de movilidad del intruso, encontramos que el menor error en la estimación se obtiene cuando el intruso se mueve acorde a un



a)



b)



c)

Figura 5.21: Valores de sobrecarga de encaminamiento para los protocolos DABR y MRLG cuando varía: a) la cantidad de tráfico inyectado, b) la velocidad del intruso, y c) el número de nodos en el escenario.

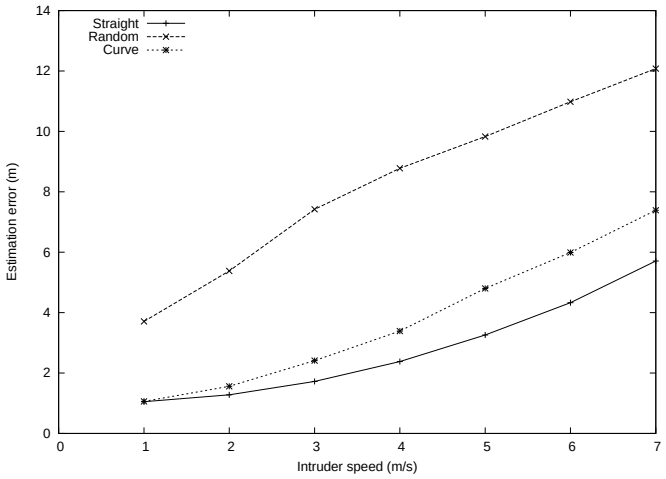


Figura 5.22: Error medio estimado para diferentes patrones de movilidad cuando varia la velocidad del intruso.

camino recto, con valores de error dentro del rango de 1 a 6 metros. Los patrones de movimiento en curva y aleatorios introducen un mayor error de estimación (de hasta 7,4 y 11,9 metros, respectivamente). Esto es esperable ya que el algoritmo de seguimiento de intrusos propuesto hace suposiciones de movimiento lineal para cada periodo de micro-grupo, tal y como se ha explicado en el capítulo anterior. Los patrones de movimiento aleatorios son los peores escenarios posibles para nuestro algoritmo de seguimiento de intrusos, lo que explica las diferencias detectadas. Sin embargo, se puede observar que, a una velocidad típica (en el intervalo [1,3] m/s), el error se mantiene razonablemente bajo, y el sistema permite que la búsqueda y seguimiento de intrusos pueda llevarse a cabo sin muchos inconvenientes en todos los casos.

5.6.2.4. Impacto de la movilidad del dreno

En esta última serie de simulaciones se analiza el comportamiento de las WSN al variar la velocidad del dreno. Debemos tener en cuenta que, en los experimentos, el dreno utilizará estimaciones de la posición del intruso a fin de avanzar continuamente hacia él, como si intentara atraparlo. Sin embargo, los niveles más altos de movilidad del dreno requieren que la topología de red deba adaptarse rápidamente. Además, el patrón de movilidad del intruso estará íntimamente relacionado con el patrón de movilidad del dreno. Por lo tanto, nuestro propósito es determinar cómo estos parámetros afectan el error de seguimiento.

Los parámetros utilizados en esta serie de simulaciones son similares a los de la sección anterior, pero se ha fijado la velocidad del intruso en 4 m/s y se ha variado la velocidad del dreno. La figura 5.23 presenta los resultados obtenidos. Esta figura

muestra que el algoritmo de encaminamiento es bastante robusto en presencia de movilidad del dreno, siendo la estimación de error para los patrones de movilidad en línea recta o curva estables, con valores de error de 5 y 6 metros (ver figura 5.23 a). En el caso de rutas aleatorias, el error estimado aumenta ligeramente (de 7 a casi 9 metros) cuando aumenta la velocidad del dreno. Para comprender esta diferencia, hay que tener en cuenta que un mayor error asociado con la estimación de la posición del intruso hace que la trayectoria seguida por el dreno también llegue a convertirse en algo más irregular, causando así más cambios en la topología.

En cuanto a la sobrecarga de encaminamiento, nos encontramos con que esta depende en mayor medida del patrón de movilidad del intruso que de la velocidad del dreno. En particular, los patrones de movilidad del intruso más irregulares (en este caso, el patrón aleatorio) se asocian con más actualizaciones de encaminamiento. De nuevo destacamos que el patrón de movilidad del dreno está íntimamente relacionado con el patrón de movilidad del intruso, lo que explica el fenómeno observado. Además, nos encontramos con que el protocolo de encaminamiento MRLG es muy eficiente en el manejo de altos niveles de movilidad del dreno, ya que la sobrecarga de encaminamiento apenas varía para velocidades altas.

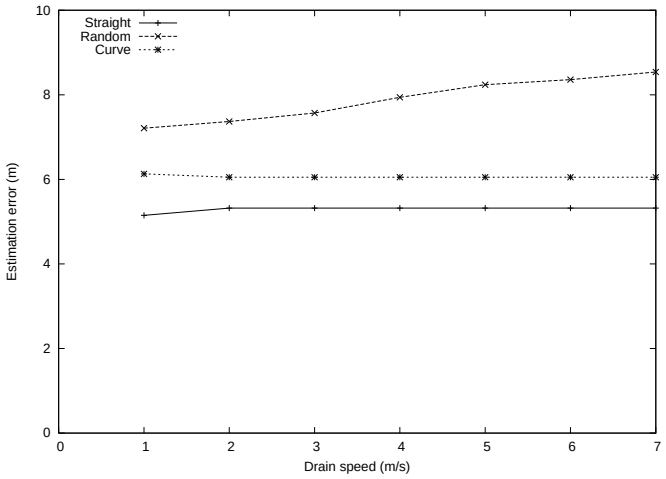
5.7. Sumario

El diseño de una WSN está influenciado por muchos factores, incluyendo restricciones de hardware, medios de transmisión, consumo de energía, topología, escalabilidad y tolerancia a fallos. La importancia de estos factores se incrementa en ambientes con varios cientos o miles de nodos sensores. Además, los protocolos y algoritmos adoptados deben ser eficientes y escalables. Cuando los objetivos son nuevas aplicaciones WSNs, como el seguimiento de intrusos con requisitos cercanos al tiempo-real, la información de las diferentes fuentes debe ser recolectada y procesada tan rápido como sea posible, para proporcionar al dreno información precisa sobre el evento en todo momento. Si, además, el dreno quiere moverse por la WSN en un intento de acercarse al intruso para capturarle, el grado de complejidad se incrementa y el protocolo de encaminamiento debe ser altamente eficiente.

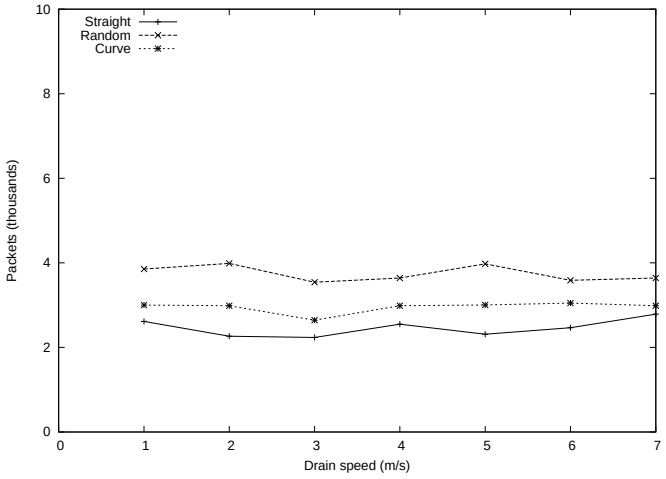
El conjunto de pruebas se realizó utilizando el simulador de redes ns-2, debido a la flexibilidad para implementar algoritmos y protocolos aplicados a las redes de sensores inalámbricas. En este simulador se integraron los dos protocolos de encaminamiento propuestos, DABR y MRLG, y se llevó a cabo una evaluación de las prestaciones de estos protocolos para aplicaciones WSNs con requisitos de tiempo real.

Como pasos preliminares a la serie de simulaciones fue necesaria la creación de escenarios para todo el conjunto de pruebas, así como realizar la configuración del fichero .tcl donde se especificaron las características de cada una de las aplicaciones a simular. Cuando ya se tuvieron los resultados de las simulaciones (ficheros traza) se utilizaron *scripts* para el análisis y generación de las gráficas de la información obtenida de estos ficheros traza.

En el estudio realizado primero se determinó la mejor tasa de actualización de rutas para el algoritmo de encaminamiento en la presencia de nodos sensores consumidos



a)



b)

Figura 5.23: Error medio estimado para diferentes patrones de movilidad (a) y sobrecarga de encaminamiento (b) al variar la velocidad del dreno.

por el fuego. Posteriormente, se evaluaron las prestaciones de ambos protocolos de encaminamiento, especificando en cada uno de ellos los parámetros de las simulaciones, auxiliados por una herramienta que nos permitió especificar el tipo de evento, así como proporcionar algunos datos de entrada como el escenario, el fichero de salida, la longitud del escenario en x y en y, la velocidad de propagación, e incluso el ángulo de propagación. Para ambos protocolos evaluados se obtuvieron la tasa de pérdida, el promedio del retardo extremo a extremo, y la sobrecarga de encaminamiento.

El análisis realizado nos permitió comparar las áreas afectadas real y estimada, y así determinar la efectividad del rastreo de eventos mediante aplicaciones WSN. Encontramos que la propagación del fuego difiere de la propagación de gas debido a la destrucción de los nodos sensores, siendo necesaria una actualización de rutas. Esto provoca pérdida de datos y retardos, e incrementa la inexactitud del proceso de rastreo. En lo que respecta al seguimiento de intrusos, los resultados experimentales muestran que, de forma general y para altos niveles de movilidad tanto del intruso como del dreno que la estrategia propuesta permite que el error de seguimiento se mantenga por debajo de 10 metros, incluso para patrones de movilidad muy irregulares. Por lo tanto, consideramos que los resultados obtenidos validan la solución propuesta.

Capítulo 6

Conclusiones

En este capítulo se presenta un resumen de las principales contribuciones y propuestas de esta tesis, así como las conclusiones de los resultados obtenidos con respecto a la utilización de WSNs para la detección y seguimiento de eventos críticos, tales como la propagación de gas y fuego, y el seguimiento de intrusos. También se presenta una descripción de cada una de las publicaciones relacionadas con la tesis. Finalmente, se plantean algunas líneas de trabajo para su desarrollo futuro.

6.1. Principales contribuciones

Como principales contribuciones de esta tesis doctoral, se pueden mencionar las siguientes:

- Análisis y desarrollo del protocolo de encaminamiento DABR para WSNs con nodos y dreno estáticos.
- Análisis y desarrollo del protocolo de encaminamiento MRLG para WSNs con nodos fijos y dreno móvil.
- Implementación de una herramienta generadora de eventos para aplicaciones de propagación de gas y fuego, así como de seguimiento y detección de intrusos.
- Integración de los dos protocolos de encaminamiento en el simulador de redes ns-2.
- Estudio, análisis y evaluación de las prestaciones ofrecidas por los protocolos DABR y MRLG en WSNs con soporte a requisitos de tiempo real suave, utilizando el estándar IEEE 802.15.4.
- Desarrollo y evaluación de algoritmos de agregación de datos y reconstrucción de eventos para poder hacer una estimación de la precisión lograda en cada uno de los eventos analizados.

6.2. Conclusiones

El estudio, análisis y evaluación realizada en el área de las WSNs con soporte a requisitos cercanos al tiempo real, nos ha permitido evaluar las prestaciones logradas con cada uno de los protocolos de encaminamiento propuestos, en los diferentes tipos de eventos modelados. El estudio realizado contempla las características de los nodos sensores en la red, las características con las que debe contar la arquitectura de comunicación bajo los requisitos del estándar IEEE 802.15.4, los protocolos de encaminamiento, y la evaluación del rendimiento, la cual ha sido realizada tanto en WSNs con drenos estáticos como móviles.

Con respecto al modelado y seguimiento de eventos críticos, se ha realizado un estudio detallado de la propagación de eventos de gas y fuego, así como el seguimiento de intrusos, incluyendo su modelado. Para el caso de propagación de gas y fuego, se ha tenido en cuenta la propagación de estos tipos de eventos tanto en interiores como en exteriores. Respecto al modelado los patrones de movilidad de intrusos, nos hemos centrado en tres tipos de patrones: movimiento recto, aleatorio y siguiendo un patrón de movimiento curvado, dejando abierta la posibilidad de incluir otros modelos matemáticos distintos. El modelado de cada uno de los eventos se ha traducido en una implementación e integración en una herramienta generadora de eventos, que tiene la finalidad de poder simular todos estos tipos de eventos mencionados, facilitando así la simulación de los eventos mediante el simulador de redes ns-2.

Como complemento a los modelos citados anteriormente, también se han desarrollado y evaluado algoritmos de agregación de datos y reconstrucción de eventos, lo que ha permitido hacer una estimación de la precisión lograda en cada uno de los eventos analizados. Finalmente, mediante la interfaz gráfica de usuario, se permite la interacción entre el usuario final y la herramienta desarrollada, permitiendo proporcionar los datos de entrada, visualizar los eventos y generar el tráfico correspondiente a todos los tipos de eventos modelados.

Se han propuesto dos protocolos de encaminamiento: el DABR, un protocolo proactivo para WSNs estáticas basado en anuncio del dreno, y el MRLG, con soporte a drenos móviles, ambos implementados en el simulador de redes ns-2. Se realizó una serie considerable de pruebas para evaluar el rendimiento en el encaminamiento con respecto a estos dos protocolos de encaminamiento. Los experimentos de simulación llevados a cabo permitieron evaluar las prestaciones del protocolo de encaminamiento DABR en escenarios de dreno estático, y el desempeño de los protocolos DABR y MRLG en escenarios con dreno dinámico. Cabe destacar que, para aplicaciones WSN con dreno estático, la arquitectura propuesta basada en el protocolo de encaminamiento DABR presentó un buen rendimiento en cuanto a las métricas evaluadas (retardo, tasa de pérdida y sobrecarga), mientras que para aplicaciones WSN con dreno móvil, y evaluando las mismas métricas, el protocolo MRLG presentó un buen rendimiento.

Mediante el uso de la herramienta generadora de eventos, también se obtuvieron las medidas de precisión de los eventos generados, tanto en la propagación de gas y fuego, como en el seguimiento de intrusos. Para llevar a cabo un análisis de estas aplicaciones en una WSN, se ha utilizado la herramienta modeladora propuesta, habiéndose realizado un estudio con el objetivo de garantizar la efectividad de una WSN

basada en el estándar IEEE 802.15.4. Encontramos que la propagación del fuego difiere de la propagación de gas debido a la destrucción de los nodos sensores, ya que los enlaces en la WSN son frecuentemente rotos, siendo necesaria una actualización de rutas, lo que provoca pérdida de datos y retardos, e incrementa la inexactitud del proceso de rastreo. A pesar de ello, el grado de error encontrado para el caso del fuego puede ser considerado adecuado para la aplicación deseada. Con respecto al seguimiento de intrusos, los resultados experimentales muestran que, de forma general, e incluso bajo altos niveles de movilidad tanto del intruso como del dreno, la estrategia propuesta permite que el error de seguimiento se mantenga por debajo de 10 metros, incluso para patrones de movilidad muy irregulares. Finalmente, se propuso reconstruir la propagación de eventos de gas y fuego usando las trazas de simulación. Esto nos permitió comparar las áreas afectadas real y estimada, y así determinar la efectividad del rastreo de este tipo de eventos mediante WSNs.

Globalmente consideramos que los objetivos de la tesis han sido alcanzados, por lo que damos por terminada esta disertación.

6.3. Publicaciones relacionadas con la tesis

El trabajo realizado para esta tesis ha generado hasta el momento las publicaciones siguientes:

- "Modeling emergency events to evaluate the performance of time-critical WSNs", Carlos T. Calafate, Carlos Lino, Juan-Carlos Cano, Pietro Manzoni, *IEEE Symposium on Computers and Communications (ISCC 2010)*, Riccione, Italy. June 22-25, 2010.

  Para evaluar con precisión el rendimiento de las WSNs para rastrear eventos en tiempo-real, en este paper se desarrolla un marco de referencia (*framework*) generador de eventos compatible con el simulador ns-2, que es capaz de modelar tanto eventos para detección de intrusos, como eventos de propagación de fuego o gas en escenarios interiores y exteriores. En este artículo hacemos una descripción analítica del *framework* desarrollado, y se presenta la herramienta propuesta junto con ejemplos visuales de diferentes tipos de eventos. Esta herramienta nos permite evaluar la efectividad de una WSN en el soporte a aplicaciones de misiones críticas mediante simulaciones.

- "Design and evaluation of a routing scheme based on drain announcements for IEEE 802.15.4 based WSNs", Carlos Lino, Carlos T. Calafate, Pietro Manzoni, Juan-Carlos Cano, Arnoldo Díaz. *XXI Jornadas de Paralelismo, CEDI 2010*, Valencia, Spain. 7-10 September, 2010.

  En este artículo se presenta el diseño y la propuesta del algoritmo de encaminamiento DABR (*Drain Announcements Based Routing*), con el cual se pretende reducir la sobrecarga de encaminamiento para el descubrimiento de rutas. El algoritmo diseñado también pretende reducir el retardo extremo a extremo al tener poco tráfico de encaminamiento en los canales de comunicación. La implementación de este protocolo de encaminamiento se basa en el anuncio del

dreno, centrándose en escenarios donde los nodos sensores y el nodo dreno son fijos, distribuidos en una topología de malla dentro de un espacio físico.

- "Efficient routing in large sensor grids supporting mobile drains", Carlos Lino, Carlos T. Calafate, Arnoldo Diaz-Ramirez, Pietro Manzoni, Juan-Carlos Cano. *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2011)*, Lucca, Italy. June 20-24, 2011.

  En este artículo se propone un nuevo protocolo de encaminamiento llamado MRLG (*Mobile-drian Routing for Large Grids*), el cual soporta movimiento del dreno dentro una WSN de forma eficiente. El algoritmo propuesto puede ser utilizado en la detección de intrusos, y está optimizado para operar conjuntamente con el estándar IEEE 802.15.4. El protocolo se caracteriza por un bajo consumo de energía, una baja latencia, y la habilidad de permitir conectar un gran número de nodos sensores (hasta 2^{16} dispositivos) en una WSN. El algoritmo de encaminamiento MRLG tiene la finalidad de soportar la movilidad de un dreno en WSNs, y ofrece prestaciones óptimas en escenarios con un gran número de nodos distribuidos en forma de malla.

- "Evaluating the performance of the IEEE 802.15.4 standard in supporting time-critical Wireless Sensor Networks", Carlos Lino, Carlos T. Calafate, Arnoldo Diaz, Juan Carlos Cano, Pietro Manzoni, *Book Chapter in "Advancements in Distributed Computing and Internet Technologies: Trends and Issues", edited by Al-Sakib Khan Pathan, Mukaddim Pathan, and Hae Young Lee. Published by IGI Global in August, 2011.* DOI: 10.4018/978-1-61350-110-8. ISBN13: 9781613501108.

  En esta publicación se lleva a cabo una evaluación del rendimiento del estándar IEEE 802.15.4, partiendo de la propuesta de un protocolo de encaminamiento mediante el cual se pretende reducir la sobrecarga de encaminamiento para el descubrimiento de rutas por los nodos sensores, los cuales detectan y requieren enviar la información a su destino. El algoritmo permite reducir el retardo extremo a extremo, ya que éste introduce poco tráfico de encaminamiento en los canales de comunicación.

- "Studying the Feasibility of IEEE 802.15.4-Based WSNs for Gas and Fire Tracking Applications Through Simulation", Carlos Lino, Carlos T. Calafate, Arnoldo Díaz-Ramirez, Juan-Carlos Cano, Pietro Manzoni, *11th IEEE International Workshop on Wireless Local Networks (WLN 2011)*, Bonn, Germany. 4-7 October 2011.

  En este artículo nos centramos en aplicaciones de WSN para monitorizar entornos de interiores y exteriores. Se propone un sistema de monitorización con prestaciones cercanas al tiempo real, en aplicaciones de detección de eventos de gas y fuego. El rendimiento de cada uno de estos eventos se evalúa usando la tecnología IEEE 802.15.4, y utilizando un esquema de encaminamiento para WSNs basado en anuncios del dreno para el descubrimiento de ruta. El protocolo de encaminamiento se ha desarrollado con el objetivo de reducir el tráfico

de control al mínimo. Para evaluar el rendimiento también se utiliza la herramienta modeladora de eventos, que permite comparar la entrada y la salida de los eventos para determinar el grado de precisión logrado en el proceso.

- "Intruder tracking in WSNs using binary detection sensors and mobile sinks", Carlos Lino, Tomás Navarro, Carlos T. Calafate, Arnoldo Díaz-Ramirez, Juan-Carlos Cano, Pietro Manzoni, IEEE Wireless Communications and Networking Conference (WCNC 2012), Paris, France. April 1-4, 2012.

  En este artículo nos centramos en la precisión del seguimiento y monitorización de intrusos, basados en mecanismos de detección binaria de bajo coste. Para superar las limitaciones impuestas por este tipo de sensores, se propone un algoritmo de seguimiento de intrusos para estimar la localización de los mismos. Se hace un estudio con el estándar IEEE 802.15.4 para las comunicaciones de radio y, se utiliza el protocolo de encaminamiento de datos con un nodo móvil llamado MRLG (Mobile-sink routing for large grids). Los resultados experimentales están basados en un despliegue de sensores en malla que muestran el error de seguimiento, considerando medidas como la distancia euclidiana media entre las ubicaciones del intruso real y estimada.

- "An Efficient Solution Offering Sink Mobility Support in Wireless Sensor Networks", Carlos Lino, Carlos T. Calafate, Arnoldo Díaz, Pietro Manzoni and Juan-Carlos Cano, 11th Wireless Telecommunications Symposium (WTS 2012), Londres, England, UK. April 18-20, 2012.

  En este artículo, nos centramos en las aplicaciones que requieren soporte para la movilidad, incluyendo escenarios para detección y persecución de intrusos. Las comunicaciones son basadas en el estándar IEEE 802.15.4 debido a su bajo consumo de energía, baja latencia y la capacidad de conectar un gran número de nodos de sensores en una WSN. En este paper se propone un novedoso algoritmo Mobile-sink Routing for Large Grid (MRLG), con el propósito de dar soporte a drenos móviles en las WSNs. MRLG permite reducir la carga de encaminamiento basándose en procesos de recuperación de rutas locales, lo que proporciona una eficiencia significativa en escenarios con un gran número de sensores. Los resultados experimentales muestran que, en comparación con el estándar de estrategias de encaminamiento basadas en anuncios del dreno, tales como el Collection Tree Protocol (CTP), el desempeño del algoritmo MRLG es superado significativamente en términos de tasa de entrega de paquetes, retardo de extremo a extremo, y sobrecarga de encaminamiento.

6.4. Trabajo futuro

Los resultados alcanzados en esta tesis doctoral suponen un avance en el estado del arte de la investigación en el área de la monitorización de entornos con requisitos de baja latencia. Consideramos que las aportaciones realizadas orecen un nuevo punto de partida dónde existe un amplio abanico de posibilidades en términos de trabajos de

investigación. En detalle, creemos que se puede dar continuidad a esta tesis mediante las siguientes líneas de trabajo:

- Incrementar la capacidad de la herramienta generadora de eventos para permitir modelar una mayor cantidad de eventos, así como mejorar la precisión de los modelos ya existentes.
- Diseñar protocolos de encaminamiento de alta eficiencia que den soporte a múltiples drenos, y que combinen requisitos de QoS, consumo energético y balanceo de carga.
- Ampliar los algoritmos propuestos para estimación de áreas afectadas por gas y fuego, y la posición de intrusos, para incluir múltiples zonas afectadas y múltiples intrusos.
- Utilizar datos provenientes de sensores más complejos, incluyendo sensores de vídeo y voz, lo que permitirá realizar la monitorización con un grado de sofisticación que va mucho más allá de lo permitido con sensores binarios, como los utilizados en esta tesis.

Como trabajo a corto plazo, planeamos implementar, desplegar y probar la solución propuesta en este trabajo utilizando un banco de pruebas real, para validar los elevados niveles de eficiencia obtenidos en esta tesis.

Bibliografía

[1] A. Chehri, P. Fortier, and P. M. Tardif, "Security monitoring using wireless sensor networks," *Communication Networks and Services Research, Annual Conference on*, vol. 0, pp. 13–17, 2007.

[2] P. Chen, S. Oh, M. Manzo, B. Sinopoli, C. Sharp, K. Whitehouse, O. Tolle, J. Jeong, P. Dutta, J. Hui, S. Schaffert, S. Kim, J. Taneja, B. Zhu, T. Roosta, M. Howard, D. Culler, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 3128–3133, June 2006.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292–2330, August 2008.

[4] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[5] G. M. A. C. Roberto Verdone, Davide Dardari, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Academic Pres, January 2008.

[6] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pp. 467–472, August 2000.

[7] J. Zheng and M. J. Lee, "A comprehensive performance study of ieee 802.15.4," *Kluwer Academic Publishers Hingham, MA, USA*, p. 14, 2003.

[8] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: Self-configuring localization systems," *Proc. 6th International Symposium on Communication Theory and Applications (ISCTA 01), Ambleside, Lake District*, p. to appear, July 2001.

[9] I. D. Chakeres and E. M. Belding-Royer, "Aodv routing protocol implementation design," *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pp. 698–703, August 2004.

[10] Z. Alliance, *ZigBee Specifications, version 1.0*, April 2005.

[11] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "Spins: Security protocols for sensor networks," *Wireless Networks*, vol. 8, pp. 189–199, September 2001.

[12] A. Hac and A. Hac, *Wireless Sensor Network Designs.* Wiley (December 17, 2003), December 2003.

[13] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer, IEEE Computer Society*, vol. 35, no. 10, pp. 54–62, 2002.

[14] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 113–127, May 2003.

[15] I. 802.15.4, *IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wirelees Personal Area Networks (WPANs).* IEEE Computer Society, ieee std 802.15.4 2006 ed., Junio 2006.

[16] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.

[17] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," *MobiHoc '03 Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 35–45, 2003.

[18] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, and M. Gerla, "Measuring effective capacity of ieee 802.15.4 beaconless mode," *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 493 –498, 3-6 2006.

[19] J. Misic, S. Shafi, and V. B. Misic, "Maintaining reliability through activity management in 802.15.4 sensor networks," *Quality of Service in Heterogeneous Wired/Wireless Networks, International Conference on*, vol. 0, p. 5, 2005.

[20] J. Misic, V. B. Misic, and S. Shafi, "Performance of ieee 802.15.4 beacon enabled pan with uplink transmissions in non-saturation mode - access delay for finite buffers," *Broadband Networks, International Conference on*, vol. 0, pp. 416–425, 2004.

[21] M. Neugebauer, J. Plonnigs, and K. Kabitzsch, "A new beacon order adaptation algorithm for ieee 802.15.4 networks," *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pp. 302 – 311, 2005.

[22] A. Koubaa, M. Alves, and E. Tovar, "Gts allocation analysis in ieee 802.15.4 for real-time wireless sensor networks," *20th IEEE International Parallel & Distributed Processing Symposium*, p. 8 pp., April 2006.

[23] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234–244, October 1994.

[24] C.-C. Chiang, "Routing in clustered multihop mobile wireless networks with fading channel," *Proceedings of IEEE SICON 97*, pp. 197–211, 1997.

[25] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mob. Netw. Appl.*, vol. 1, pp. 183–197, Oct 1996.

[26] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference, IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pp. 62 – 68, 2001.

[27] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector (aodv) routing," *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA ́99. Second IEEE Workshop on*, pp. 90 – 100, 1999.

[28] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," pp. 153–181, 1996.

[29] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *In Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 97), ACM Press*, 1997.

[30] W. Heinzelman, J. Kulik, and H. Balakrishnan, "In adaptative protocols for information dissemination in wireless sensor networks," 1999.

[31] A. Manjeshwar and D. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks.," *In Parallel and Distributed Processing Symposium, Proceedings 15th International*, pp. 2009–2015, 2001.

[32] R. Fonseca, K. K. S. Gnawali, O.and Jamieson, and A. Levis, P.and Woo, "The collection tree protocol.," *University of Berkeley*, vol. 123, 2007.

[33] TinyOS, "http://www.tinyos.net,"

[34] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," *ACM New York, NY*, vol. 5, pp. 96 – 108, December 2003.

[35] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks," *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*, pp. 49 – 58, agosto 2002.

[36] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *ACM, MOBICOM'2000*, pp. 56–67, 2000.

[37] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *Appearing in 5th Annual Symposium on Operating Systems Design and Implementation (OSDI).*, vol. 36, December 2002.

[38] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.

[39] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, pp. 11–25, October 2001.

[40] "Force xxi battle command, brigade-and-below," tech. rep., http://www.fas.org/man/dod-101/sys/land/fbcb2.htm, 1998.

[41] A. Ledeczi, A. Nadas, P. Volgyesi, G. Balogh, B. Kusy, J. Sallai, G. Pap, S. Dora, K. Molnar, M. Maroti, and G. Simon, "Countersniper system for urban warfare," *ACM Transactions on Sensor Networks*, vol. 1, pp. 153–177, November 2005.

[42] "Self-healing minefield," tech. rep., http://www.globalsecurity.org/military/ systems/munitions/shm.htm, 2003.

[43] J. Agre and L. Clare, "An integrated architecture for cooperative sensing networks," *IEEE Computer Magazine*, vol. 33, pp. 106–108, May 2000.

[44] M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, "Upper bounds on the lifetime of sensor networks," *IEEE International Conference on Communications ICC'01*, vol. 3, pp. 785 – 790, June 2001.

[45] P. Bonnet and P. S. Gehrke, "Querying the physical world," *IEEE Personal Communications*, vol. 7, pp. 10–15, October 2000.

[46] J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile network for smart dust," *Proceedings of the ACM MobiCom*, pp. 271–278, Washington 1999.

[47] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring behavior in home using a smart fall sensor," *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, pp. 607–610, Oct 2000.

[48] C. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, N. Burkhart, and A. Minassians, "Wireless sensor networks for home health care," *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 2, pp. 832 – 837, Ontario 2007.

[49] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," *ACM SIGCOMM - Latin America & Caribbean 2000, San Jose, Costa Rica*, vol. 1, April 2001.

[50] G. Bernat, A. Burns, and A. Llamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, pp. 308–321, April 2001.

[51] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," *WSNA '02 Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 88–97, 2002.

[52] E-SENSE, "Capturing ambient intelligence for mobile communications through wireless sensor networks," tech. rep., http://www.ist-esense.org/, 2007.

[53] J. Zheng and M. J. Lee, "Will ieee 802.15.4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard," *Communications Magazine, IEEE*, vol. 42, pp. 140–146, June 2004.

[54] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "Rap: a real-time communication architecture for large-scale wireless sensor networks," *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, pp. 55–66, January 2003.

[55] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. Abdelzaher, "Achieving real-time target tracking using wireless sensor networks," *In IEEE RTAS 2006*, December 2006.

[56] X.-Z. Lin, J.-J. Zhou, and C.-D. Mu, "Collective real-time qos in wireless sensor networks," *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on*, pp. 1–4, September 2006.

[57] L. Klingbeil and T. Wark, "A wireless sensor network for Real-Time indoor localisation and motion monitoring," *IPSN 08 Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 39–50, April 2008.

[58] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B: Chemical (2005) Elsevier B.V.*, vol. 110, pp. 304–311, Febrero 2005.

[59] X. Wang, J. Lizier, O. Obst, M. Prokopenko, and P. Wang, "Spatiotemporal anomaly detection in gas monitoring sensor networks," *Springer-Verlag Berlin, Heidelberg*, vol. 4913, pp. 90–105, 2008.

[60] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, vol. 2, pp. 1214–1217, 2005.

[61] J. Zhang, W. Li, N. Han, and J. Kan, "Forest fire detection system based on a zigbee wireless sensor network," *Frontiers of Forestry in China*, vol. 3, no. 3, pp. 369–374, 2008.

[62] B. Son, Y. sork Her, and J.-G. Kim, "A design and implementation of forest-fires surveillance system based on wireless sensor networks for south korea mountains," *International Journal of Computer Science and Network Security*, vol. 6, pp. 124–130, September 2006.

[63] Y. H. H. Dan Li, Kerry D. Wong and A. M. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 17–29, 2002.

[64] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks, Elsevier*, vol. 46, pp. 605–634, December 2004.

[65] Q. Cao, T. Yan, J. Stankovic, and T. Abdelzaher, "Analysis of target detection performance for wireless sensor networks," *In DCOSS 05*, pp. 276–292, 2005.

[66] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," *ACM Press, In Mobisys*, pp. 270–283, 2004.

[67] R. G. Rehm, "The effects of winds from burning structures on ground-fire propagation at the wildland-urban interface," *EPEW'10 Proceedings of the 7th European performance engineering conference on Computer performance engineering*, vol. 12, pp. 477–496, Junio 2008.

[68] F. Grinstein, L. G. Margolin, W. Rider, and O. Parmhed, *Implicit large eddy simulation: computing turbulent fluid dynamics*, ch. 17, p. 543. Cambridge University Press, 2007.

[69] B. Abdalhaq, *A methodology to enhance the prediction of forest fire propagation.* PhD thesis, Universitat Autònoma de Barcelona, España, Junio 2004.

[70] T. Cam, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *WIRELESS COMMUNICATIONS & MOBILE COMPUTING (WCMC): SPECIAL ISSUE ON MOBILE AD HOC NETWORKING: RESEARCH, TRENDS AND APPLICATIONS*, vol. 2, pp. 483–502, 2002.

[71] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 99–108, April 2003.

[72] "The network simulator, ns-2." http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[73] "Gnuplot." http://www.gnuplot.info/.

[74] Y. Li, M. T. Thai, and W. Wu, "Wireless sensor networks and applications li, y. 978-0-387-49592-7," *Boston, MA : Springer Science+Business Media, LLC*, 2008.

[75] C. Lino, C. T. Calafate, A. Diaz, P. Manzoni, and J.-C. Cano, *Advancements in Distributed Computing and Internet Technologies: Trends and Issues*, ch. Evaluating the performance of the IEEE 802.15.4 standard in supporting time-critical Wireless Sensor Networks, pp. 142–158. IGI Global, 2011.

[76] C. Lino, C. T. Calafate, A. Diaz, P. Manzoni, and J.-C. Cano, "Efficient routing in large sensor grids supporting mobile drains," *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, (WoWMoM'2011), Lucca, Italy*, pp. 1–3, June 20-24 2011.

[77] "J-sim.." http://sites.google.com/site/jsimofficial/.

[78] R. Barr, Z. J. Haas, and R. van Renesse, "Jist: Embedding simulation time into a virtual machine," *IN EUROSIM CONGRESS ON MODELLING AND SIMULATION*, p. 16, 2003.

[79] "Nctuns 2.0 network simulator and emulator.." http://nsl.csie.nctu.edu.tw/nctuns.html.

[80] "Omnet++ discrete event simulator.." http://www.omnetpp.org.

[81] "Ptolemy ii. heterogeneous model and design.." http://ptolemy.eecs.berkeley.edu/ptolemyII.

[82] L. F. Perrone and D. M. Nicol, "A scalable simulator for tinyos applications," *Simulation Conference, 2002. Proceedings of the Winter*, pp. 679–687, December 2002.

[83] C. Lino, C. T. Calafate, A. Diaz, P. Manzoni, and J.-C. Cano, "Design and evaluation of a routing scheme based on drain announcements for ieee 802.15.4 based wsns," *XXI Jornadas de Paralelismo, CEDI 2010, Valencia, Spain*, vol. 1, pp. 855–862, 7-10 September 2010.