TESIS DEFENDIDA POR

Luis Enrique Palafox Maestre

Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. José Antonio García Macías
Director del Comité

Dr. Jesús Favela Vara
Miembro del Comité

Dr. Javier Gómez Castellanos
Miembro del Comité

Dr. José Alberto Fernández Zepeda
Miembro del Comité

Dr. Luis Armando Villaseñor González
Miembro del Comité

Dra. Ana Isabel Martínez García
Coordinadora del programa de posgrado en Ciencias de la Computación

Dr. David Hilario Covarrubias Rosales
Director de Estudios de Posgrado

19 de noviembre de 2009

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

PROGRAMA DE POSGRADO EN CIENCIAS EN CIENCIAS DE LA COMPUTACIÓN

DISEMINACIÓN DE DATOS EN REDES INALÁMBRICAS DE SENSORES REDUNDANTES

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

DOCTOR EN CIENCIAS

Presenta:

LUIS ENRIQUE PALAFOX MAESTRE

Ensenada, Baja California, México, noviembre de 2009

RESUMEN de la tesis de **LUIS ENRIQUE PALAFOX MAESTRE**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN . Ensenada, Baja California, noviembre de 2009.

DISEMINACIÓN DE DATOS EN REDES INALÁMBRICAS DE SENSORES REDUNDANTES

Resumen aprobado por:

Dr. José Antonio García Macías

Director de Tesis

En este trabajo, se propone un protocolo de diseminación de datos de nodo de sensado a nodo líder en Redes Inalámbricas de Sensores (*WSN*, por sus siglas en inglés) estructuradas jerárquicamente. El protocolo propuesto se basa en la detección oportuna de datos redundantes. Esta detección se hace mediante la comparación del Código de Autentificación de Mensaje (código *MAC*) de los diferentes nodos del cluster. El hecho de que se utilice el código MAC hace posible la integración de servicios básicos de seguridad como valor agregado a este trabajo, esto sin incurrir en un consumo adicional de energía. Con la finalidad de validar la efectividad del protocolo propuesto, se implementaron dos aplicaciones de WSN, la primera que consiste en un sistema de monitoreo ambiental encargado de sensar temperatura, humedad, luz infrarroja y luz visible; y la segunda aplicación de captura de voz con WSN orientada a ambientes de cómputo ubicuo en el hogar. A través varios experimentos se demostró que la propuesta presentada extiende en ambas aplicaciones el tiempo de vida de la red cuando se presentan niveles de redundancia mayores a la mitad del número de nodos que conforman el cluster. Concretamente, la contribución de este trabajo es una nueva alternativa eficiente para WSN jerárquicas altamente pobladas, las cuales en la mayoría de los casos presentan altos níveles de redundancia de datos.

Palabras Clave: redes inalámbricas de sensores, redundancia de datos, seguridad en redes inalámbricas de sensores, redes de sensores de audio, diseminación de datos.

ABSTRACT of the thesis presented by **LUIS ENRIQUE PALAFOX MAESTRE**, in partial fulfillment of the requirements of the DOCTOR IN SCIENCE degree in COMPUTER SCIENCE . Ensenada, Baja California, november 2009.

DATA DISSEMINATION IN REDUNDANTLY DEPLOYED WIRELESS SENSOR NETWORKS

In this work, a member-node to cluster-head data dissemination protocol for hierarchical Wireless Sensor Networks (WSN) is proposed. The proposed protocol is based in the early detection of data redundancy. This detection is done through comparing the Message Authentication Code (*MAC* code) belonging to the different nodes in the cluster. The fact that the MAC is being used, enables the integration of basic security services without incurring in additional power consumption. In order to validate the effectiveness of the proposed protocol, two WSN applications were implemented, the first one is an environmental monitoring system which senses temperature, humidity, infrared and visible light; the other application is an audio capture WSN oriented for home ubicomp environments. Through several experiments, it was demonstrated that network lifetime is extended when the redundancy level exceeds half of the number of member nodes in the cluster. Summarizing, the contribution of this work is a novel and efficient alternative for densely populated hierarchical WSN, in which most of the times, high levels of data redundancy appear.

Keywords: wireless sensor networks, data redundancy, security in wireless sensor networks, data dissemination.

A Luisingui y Andy...

AGRADECIMIENTOS

Mis más sinceros agradecimientos a todos los que de una manera u otra hicieron que este sueño se hiciera realidad.

A mi esposa Gabby, por darme todo el apoyo que necesito, sobre todo en los momentos más díficiles.

A mis hijos Luis y Andrés, por recordarme cada mañana lo que es realmente importante en esta vida.

A mi mamá, por ser mi más grande ejemplo a seguir.

A mi abuelita Mina, quien siempre me ha enseñado que se puede ser fuerte y perseverante sin dejar de ser sensible.

A mi papá y a mis hermanos Jorge, Michel y Genaro; por su gran apoyo.

A mis suegros, por darme siempre su apoyo incondicional con todo y que me llevé a los más preciado de su vida.

Al "Dr. Tony", por ser un excelente guía en el medio académico, pero más importante aún, por su amistad y su inagotable paciencia.

A los miembros del comité de tesis, por sus invaluables consejos.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo económico.

A mis amigos de CICESE: Edgardo, Lupita, Leonardo Galicia, Argelia, Ismael, Leonardo Trujillo; gracias por traer una sonrisa, especialmente en esos momentos cuando las cosas no salen como estaban planeadas.

A mis compañeros y amigos de la Facultad de Ciencias Químicas e Ingeniería, especialmente a la M.C. Ma. Eugenia Pérez Morales y al M.C. Rubén Sepúlveda por el gran apoyo brindado durante su gestión.

Pero sobre todo a Dios, por rodearme de personas como las que ya mencioné y otras que seguramente olvido...mi mente puede que olvide, pero mi corazón nunca lo hará.

CONTENIDO

Página

Contenido (continuación)

Página

Contenido (continuación)

Página

LISTA DE FIGURAS

Lista de Figuras (continuación)

LISTA DE TABLAS

Tabla Página

Capítulo I

Introducción

I.1 Redes Inalámbricas de Sensores

I.1.1 Definición general

Las Redes Inalámbricas de Sensores (WSN por sus siglas en inglés) están formadas por pequeños dispositivos que cuentan con capacidad de comunicación inalámbrica y con sensores específicos a la aplicación para la cual se van a utilizar. Típicamente, este tipo de redes se instalan con el fin de monitorear algún fenómeno en particular, por ejemplo: detección oportuna de incendios forestales, monitoreo de habitat de alguna especie animal, monitoreo de la contaminación ambiental, entre muchas otras. El contar con capacidad de comunicación aunado al hecho de que se han venido desarrollando protocolos de comunicación multisaltos, han permitido que el área que pueden monitorear este tipo de redes en su conjunto se extienda al orden de kilómetros. A pesar de que esta tecnología ha abierto la puerta al desarrollo de aplicaciones como las que ya se mencionaron, también ha presentado retos importantes principalmente relacionados con las extremas limitaciones de recursos de procesamiento, ancho de banda y fuente de energía. Debido a lo anterior, el uso de recursos en estas plataformas debe optimizarse. Existen diversas acciones genéricas que pueden adoptarse con tal fin: mantener encendidos los nodos solo cuando sea necesario, optimizar el código de los programas ejecutados en los

nodos, y sobre todo, mantener el número de operaciones de comunicación al mínimo, es decir, evitar la transmisión de datos innecesarios. En forma general, esto último fue el motivo principal de estudio de la presente tesis.

I.1.2 Historia

La historia del desarrollo de las redes inalámbricas de sensores data desde 1998 en el proyecto Smartdust (Warneke *et al.*, 2001). Uno de los objetivos de este proyecto fue el de crear un sistema de sensado autónomo con comunicación en nodos de un milímetro cúbico (ver Figura 1). Aunque este proyecto finalizó hace tiempo, ha sido el parteaguas para el surgimiento de nuevos proyectos de investigación en el área. Estos nuevos proyectos se han desarrollado en grandes centros de investigación como el NEST[1] de la Universidad de California en Berkeley y el CENS[2] de la Universidad de California en Los Angeles. Los investigadores involucrados en estos proyectos introdujeron el término *mote* para referirse a un nodo de sensado. Los nodos de sensado no han incrementado su capacidad de energía como se esperaría de acuerdo a la Ley de Moore. Típicamente, estos nodos cuentan con capacidad limitada de procesamiento y almacenamiento en comparación a las computadoras personales. Esto se puede atribuir al bajo volumen del mercado actual para este tipo de nodos, además del hecho de que utilizan microcontroladores de bajo consumo de energía.

I.1.3 Aplicaciones de WSN

Existen muchas aplicaciones para WSN en diversos ámbitos. La mayoría de ellas involucran algún tipo de monitoreo, rastreo y control. Entre esas aplicaciones se pueden

[1]Sitio web: http://webs.cs.berkeley.edu/, visitado el 24 de octubre del 2009.

[2]Sitio web: http://research.cens.ucla.edu/, visitado el 24 de octubre del 2009.

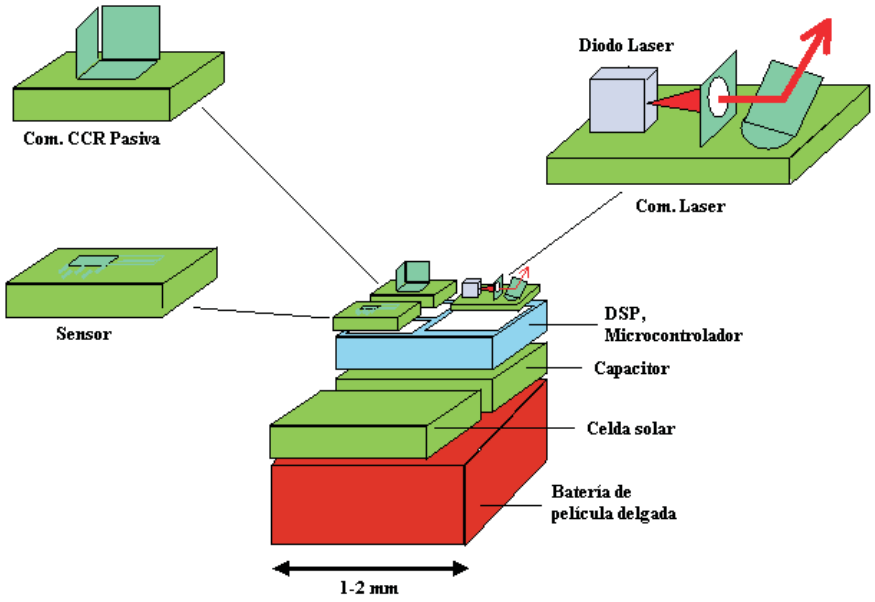

Figura 1. Componentes de un micro-nodo Smart Dust.

encontrar monitoreo de habitat, rastreo de objetos, control de reactores nucleares, detección de incendios y monitoreo de tráfico. En una aplicación típica, una WSN está dispersa en la región en la cual se pretende que recolecte datos a través de sus nodos de sensado.

Monitoreo de áreas

El monitoreo de áreas es una aplicación común para WSN. En el monitoreo de áreas, la WSN se instala en una región donde se va a monitorear algún fenómeno. Por ejemplo, una gran cantidad de nodos se pueden colocar sobre un campo de batalla para detectar la presencia de enemigos en lugar de utilizar minas (Akkaya y Younis, 2005). Cuando los sensores detectan el evento que se está monitoreando (temperatura, presión, sonido, luz, campo electromagnético, vibración, etc), el evento se requiere reportar a una de las estaciones base, la cual puede llevar a cabo una acción determinada (enviar un mensaje

por Internet o por satélite).

Monitoreo ambiental

Recientemente se han desarrollado varias aplicaciones de WSN para monitoreo ambiental. Muchas de ellas han tenido un tiempo de permanencia relativamente corto, esto debido a que las aplicaciones se desarrollaron como prototipos experimentales. Un ejemplo de aplicación de monitoreo ambiental con WSN de mayor tiempo de permanencia es la de monitoreo de glaciares (Martínez *et al.*, 2009).

Monitoreo de plantas de tratamiento de aguas residuales

Existen varias oportunidades de desarrollo de aplicaciones de redes inalámbricas de sensores dentro de la industria del tratamiento de aguas residuales. Las plantas que normalmente no cuentan con la infraestructura de instalaciones eléctricas ni de transmisión de datos (por cable) se pueden monitorear utilizando dispositivos inalámbricos y sensores alimentados por baterías o por paneles solares.

Agricultura

La utilización de redes inalámbricas de sensores en la industria agrícola ha ido creciendo considerablemente en los últimos años. Los sistemas de riego automatizados pueden monitorear el nivel en tanques de agua, las bombas de agua se pueden controlar con dispositivos de Entrada/Salida con interfaz inalámbrica, y el agua se puede medir y dicha medición enviar por medios inalámbricos a la central de cobro. Los sistemas de irrigación automática facilitan el uso eficiente de agua y reducen el desperdicio de la misma.

Las redes inalámbricas de sensores también se utilizan para controlar los niveles de

temperatura y humedad dentro de invernaderos. Cuando la temperatura y humedad baja de cierto nivel específico, al encargado del invernadero se le debe de notificar por correo electrónico o por mensaje de texto, o los sistemas automatizados deben de activar sistemas de riego, abrir ventilas, encender ventiladores o controlar una amplia gama de repuestas del sistema. Debido a que algunas redes inalámbricas de sensores son fáciles de instalar, resulta fácil moverlas según varíen las necesidades de la aplicación.

I.2 El papel de las Redes Inalámbricas de Sensores dentro del Cómputo Ubicuo

Las aplicaciones de redes inalámbricas de sensores han sido tradicionalmente de monitoreo y recolección de datos para diversos dominios tales como el de la milicia y la agricultura (Burrell *et al.*, 2004; Simon *et al.*, 2004). Por otra parte, el cómputo ubicuo hace extensivo el uso de sensores para obtener información contextual importante para que las aplicaciones puedan ajustar su comportamiento basándose en lo que el usuario pudiera requerir. Estas aplicaciones proactivas se diseñan para interrumpir al usuario lo menos posible mientras lo asisten a lo largo del día (Tennenhouse, 2000). En este trabajo se visualizan a las redes de sensores como un conjunto de nodos que están ya sea en una persona o en el ambiente. A medida de que la persona se desplaza, la información obtenida por su red de sensores personal se puede transmitir a las redes de sensores que la persona encuentre en su trayecto. Aún se requieren nuevos tipos de nodos de sensado para desarrollar aplicaciones de cómputo ubicuo más apropiadas. Estos nuevos nodos podrían incorporar nuevas formas de visualizar la información proporcionada por los sensores ubicados en el ambiente, también podrían integrar nuevas formas de

proveer datos a sensores locales y la utilización de sensores personales en conjunción con sensores fijos en el ambiente. Por otra parte, se tiene el problema de cómo generar interacciones más dinámicas entre las aplicaciones de cómputo ubicuo y el dominio de las redes inalámbricas de sensores y cómo estas interacciones impactarían los esquemas de ahorro de energía, la transferencia de datos y los protocolos de enrutamiento.

I.3 Planteamiento del problema

Las aplicaciones de redes inalámbricas de sensores pueden llegar a estar formadas por cientos o miles de nodos de sensado. Pero más importante aún es el hecho de que este tipo de nodos típicamente son muy susceptibles a fallas, debido a que se diseñaron con la idea de mantener un costo bajo de producción. Por tal motivo, en aplicaciones donde se requiera cierta confiabilidad, es necesario tomar medidas adicionales para subsanar las posibles fallas que pudieran presentar los nodos. Una de las prácticas más comúnes para lograr este objetivo, es la de colocar varios nodos de sensado cubriendo una misma área en forma redundante. De esta manera, si uno de los nodos falla, se tienen nodos adicionales como respaldo. Sin embargo, el colocar varios nodos cubriendo la misma área presenta un nuevo problema: este conjunto de nodos sensa la misma área, lo cual a su vez genera información redundante. Al transmitir dicha información, se incurre en un consumo excesivo de recursos al introducirse tráfico adicional en la red (consumiendo ancho de banda) y se gasta energía de los nodos por operaciones de comunicación mediante la unidad de radiofrecuencia. Por otra parte, cuando se habla de redes inalámbricas de sensores densamente pobladas como en el caso de las redes redundantes que se han mencionado, típicamente se adoptan topologías jerárquicas, donde cada uno de los nodos que monitorean un área de cobertura en común forman

un grupo (cluster), y tienen comunicación directa con un nodo líder, el cual coordina las actividades de los nodos que pertenecen al cluster. Entre las aplicaciones que hacen uso de este tipo de topología se puede mencionar a los sistemas de detección oportuna de incendios forestales, donde por un lado, en la mayoría de los casos, los nodos que monitorean un área específica reportan siempre las mismas condiciones normales, pero en ciertas situaciones excepcionales, reportan cambios críticos en el ambiente (por ejemplo: un incremento súbito en la temperatura). En esta aplicación, es necesario contar con una red de nodos redundantes para aumentar la confiabilidad en el sistema, y por otro lado es altamente recomendable utilizar una topología jerárquica para permitir mayor escalabilidad en la red, lo cual es un requerimiento importante en redes densamente pobladas. En general, cualquier otra aplicación que requiera un alto nivel de confiabilidad, sería también una aplicación candidato a utilizarse como caso de estudio del presente trabajo de tesis.

I.4 Objetivo general

El objetivo general del presente trabajo de tesis es el de diseñar nuevos esquemas de diseminación de datos desde nodos de sensado hacia un nodo líder, en redes inalámbricas de sensores estructuradas jerárquicamente, donde además se han colocado los nodos de sensado en forma redundante con el fin de aumentar la confiabilidad global de la red.

I.5 Objetivos específicos

- Explorar las posibles ventajas de utilizar información redundante generada en redes inalámbricas de sensores.

- Proponer nuevos esquemas de diseminación de datos en WSN jerárquicas que presentan información redundante.
- Integrar funciones básicas de seguridad a los esquemas propuestos con el fin de aplicarlos posteriormente a escenarios más demandantes.
- Implementar prototipos de los esquemas propuestos para validar su viabilidad práctica.
- Evaluar los esquemas de diseminación de datos con el fin de medir su eficiencia.
- Analizar las posibles aplicaciones de los esquemas de diseminación propuestos reconociendo las limitaciones de los mismos.

I.6 Panorámica de la tesis

Las actividades más relevantes que se llevaron a cabo durante el desarrollo del presente trabajo de tesis fueron las siguientes:

- Se realizó un estudio del estado del arte sobre la investigación en redes inalámbricas de sensores dando un énfasis en seguridad.
- Se propuso el diseño de un protocolo de diseminación de datos segura para redes inalámbricas de sensores con la idea de aprovechar los altos niveles de redundancia que se presentan en redes densamente pobladas.
- Se construyó un prototipo experimental para evaluar el desempeño del esquema propuesto.

- Se evaluó el prototipo mediante la realización de una serie de experimentos, a partir de los cuales se estimó tiempo de vida de la red, se midió la cantidad de tráfico introducido por el esquema propuesto y se determinó el impacto general de la propuesta.

- Se desarrolló un prototipo de aplicación de captura de voz utilizando redes inalámbricas de sensores, en dicho prototipo se utilizó una variante del esquema propuesto inicialmente.

- Se desarrolló la evaluación similar a la realizada al prototipo inicial.

I.7 Metodología

Durante la realización de este trabajo de tesis se ha llevado a cabo un estudio detallado del estado del arte en redes inalámbricas de sensores, cabe señalar que esta tecnología aún se puede considerar que es relativamente nueva, pues tiene escasos 10 anõs de haberse introducido en el ámbito experimental. En el estado del arte se consideraron tres vertientes principales dentro de la investigación en WSN: i) generalidades de WSN, ii) diseminación de datos redundantes en WSN y iii) seguridad en WSN; estos tres temas de investigación están relacionados con el objetivo del presente trabajo de tesis, por lo cual fue necesario considerarlos e irlos siguiendo durante la elaboración del presente trabajo de investigación. Una vez que se tuvo un panorama general del estado del arte en el área de WSN se procedió a proponer un esquema de diseminación de datos redundantes en WSN; para realizar dicha propuesta se consideraron las aplicaciones de WSN que se han publicado en la literatura, así como los esquemas de diseminación de datos redundantes, los cuales, cabe señalar que no son muchos, como se podrá

observar posteriormente en el presente documento. Una vez presentada la propuesta de diseminación de datos, se implementó en la plataforma física con la finalidad de evaluar su desempeño. Además se propuso una variante del esquema de diseminación propuesto como una aplicación que podría utilizarse potencialmente en escenarios reales de automatización de ambientes de cómputo ubicuo en el hogar mediante comandos de voz. Finalmente, se realizaron una serie de experimentos a los prototipos implementados con el fin de determinar si la propuesta tecnológica es atractiva para utilizarse a nivel de aplicación y se analizaron los resultados obtenidos.

I.8 Resultados obtenidos

Tras realizar los experimentos en los dos diferentes prototipos experimentales, se muestra que mediante la utilización del esquema propuesto se logra extender el tiempo de vida promedio de la red hasta un 17% en el nodo líder en los casos donde se presenta alto nivel de redundancia de datos comparado con esquemas tradicionales de "captura-envío". Por otra parte, se cuenta con un prototipo inicial de aplicación de captura de voz mediante redes inalámbricas de sensores, orientada a brindar soporte a personas de la tercera edad y/o con movilidad restringida.

I.9 Estructura del documento de tesis

El presente documento de tesis está estructurado de la siguiente manera: en el Capítulo II se presentan generalidades sobre WSN, entre éstas se pueden encontrar conceptos básicos, topología general, componentes de los nodos de sensado, tecnologías existentes, pila de protocolos de las WSN, limitantes de la plataforma y otras características gen-

erales; en el Capítulo III se presentan esquemas de diseminación de datos basados en el contenido de la información, entre ellos, los que se basan en la existencia de información redundante, los cuales son los de mayor interés para este trabajo de tesis; enseguida, en el Capítulo IV se presenta un enfoque alternativo para la diseminación de datos basada en algoritmos bio-inspirados; a continuación, en el Capítulo V se presentan las propuestas de protocolos de diseminación de datos basados en información redundante; en el Capítulo VI se presentan los prototipos implementados basados en las propuestas presentadas, así como los experimentos desarrollados y la presentación de los resultados. En el Capítulo VII se concluye enlistando las contribuciones del presente trabajo y describiendo posibles directrices para trabajo futuro. Adicionalmente, se presenta en el Apéndice A, un estudio sobre el trabajo relacionado con la seguridad en redes inalámbricas de sensores, esto debido, a que la integración de funciones de seguridad en las propuestas de protocolos se considera como una contribución adicional al trabajo. Finalmente en el Apéndice B se presentan las especificaciones técnicas de los nodos MicaZ, los cuales se utilizaron en una de las implementaciones que se describen en este trabajo de tesis.

Capítulo II

Características de Redes Inalámbricas de Sensores

II.1 Antecedentes

Los avances en el área de las comunicaciones inalámbricas y la electrónica han dado pie al desarrollo de pequeños nodos de bajo costo que cuentan con capacidades de sensado (Akyildiz *et al.*, 2002). Estos nodos se pueden comunicar a distancias cortas mediante el medio inalámbrico y su arquitectura básica está conformada por componentes de sensado, procesamiento de datos y comunicaciones. Este último componente hace posible que un gran número de nodos (se prevé que puedan ser miles o inclusive millones de ellos) (Warneke *et al.*, 2001) puedan integrar redes inalámbricas para comunicarse entre sí.

Una red de sensores está compuesta por un gran número de nodos que típicamente se instalan en forma masiva en un área geográfica determinada con el fin de monitorear un fenómeno. El fenómeno a observar debe de estar dentro de dicha área, la cual es conocida como *campo de sensores*. Los sensores ubicados dentro del campo de sensores comunican sus lecturas a un dispositivo (*sink*) encargado de actuar como interfaz entre la red de sensores y una red externa (típicamente el *Internet*). A través de la red externa, el usuario puede interactuar con los datos obtenidos al sensar el fenómeno. En la Figura 2 se muestra gráficamente este escenario.

Se prevé que en un futuro, en la mayoría de los casos, no se requiera predeterminar

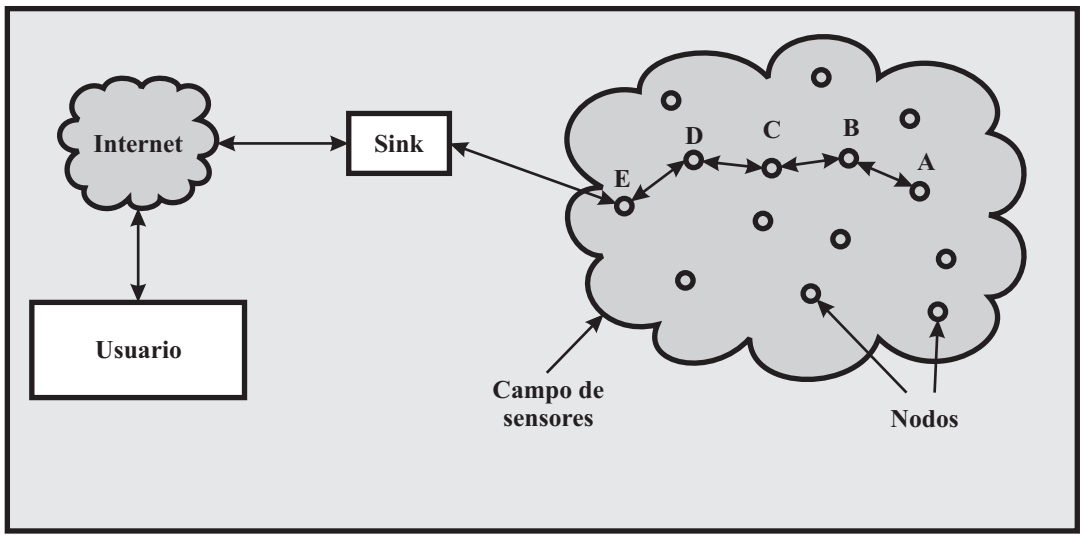

Figura 2. Escenario típico de una red de sensores.

con exactitud la ubicación exacta que tomarán los nodos al instalarse (e.g. al lanzar los nodos de una aeronave) (Römer y Mattern, 2004). Por tal motivo, este tipo de redes debe ser capaz de autoconfigurarse de acuerdo al entorno que se le presente.

Se puede ver a las redes inalámbricas de sensores como un caso especial de redes ad hoc ya que comparten algunas características de estas últimas, entre las cuales se pueden mencionar la auto-configuración y el comportamiento local. Sin embargo, existen diferencias notables entre ambos tipos de redes (García-Macías y Gómez, 2007):

- **Densidad**. De acuerdo a la visión de las redes inalámbricas de sensores, éstas contarán con miles o millones de nodos. Es decir, varios órdenes de magnitud más que las redes ad hoc.

- **Bajo costo de los nodos**. No obstante el avance tecnológico, la tendencia de los nodos en las redes de sensores es la de mantenerlos simples, es decir, incluir solo la funcionalidad básica necesaria, esto con el fin de mantener bajo los costos de los nodos y con esto hacer posible la integración de redes densas como se menciona en el punto anterior.

- **Limitantes de recursos**. Aunque las redes ad hoc presentan ciertas limitaciones, las redes de sensores llevan estas limitaciones al extremo como se menciona posteriormente en este capítulo.

- **Movilidad**. Las aplicaciones para redes ad hoc típicamente requieren que sus nodos sean móviles (*MANET, Mobile Ad hoc Networks*), en el caso de las redes inalámbricas de sensores la gran mayoría de ellas no requieren movilidad en sus nodos, aunque hay algunas excepciones (Sibley *et al.*, 2002; Kaiser *et al.*, 2003).

En este capítulo se presenta la arquitectura general de las redes de sensores así como la tecnología que existe actualmente y que ya ha hecho posible la implementación de diversas aplicaciones para resolver problemas reales.

II.2 Arquitectura general

Las redes de sensores a diferencia de las redes tradicionales de comunicaciones presentan características muy peculiares, las cuales por una parte presentan nuevos retos en diversas áreas tales como diseño de hardware, diseño de protocolos y privacidad de la información, entre otras. Pero por otra parte, hacen posible la implementación de aplicaciones de monitoreo (y en algunos casos hasta de control) distribuidas en diversos sectores, entre éstos se pueden mencionar: agricultura, medicina, militar, seguridad civil, zoología, oceanografía, geología, automotriz y otros más.

En esta sección se describen las características más sobresalientes que este tipo de redes debe reunir para cumplir con el tipo de tareas que se prevé puedan llevar a cabo.

II.2.1 Escalabilidad

El número de sensores necesario para analizar un fenómeno puede estar en el orden de los cientos de miles. Inclusive, se prevé que en un futuro, dependiendo de la naturaleza de la aplicación, este número pudiera alcanzar varios millones de nodos. Los esquemas que se propongan en un futuro deberán ser capaces de trabajar con estas grandes cantidades de nodos, de tal manera que mientras se incremente el número de nodos el desempeño general de la red no se vea afectado. Estos esquemas también deben de tomar ventaja de la alta densidad con que cuentan estas redes. La densidad puede variar desde solo algunos nodos hasta cientos de nodos por una región, la cual pudiera ser de menos de 10 metros de diámetro.

II.2.2 Tolerancia a fallas

Debido a las características del ambiente donde se instalan las redes de sensores y debido a las limitaciones que éstas presentan, sus nodos pueden fallar en cualquier momento. Las posibles fallas que se presenten en los nodos no deben de afectar al funcionamiento general de la red. La tolerancia a fallas se puede definir como la capacidad de la red de mantener su funcionamiento a pesar de la falla de algunos nodos (Hoblos *et al.*, 2000; Shen *et al.*, 2001). La tolerancia a fallas $R_k(t)$ de un nodo se modela con la distribución de Poisson (ver Ecuación 1) para determinar la probabilidad de que no se tenga una falla en un intervalo de tiempo de 0 a t (Hoblos *et al.*, 2000):

$$R_k(t) = e^{-\lambda_k t} \tag{1}$$

donde λ_k es la tasa de falla del nodo k y t es el período de tiempo.

II.2.3 Bajo costo

Debido a que las redes de sensores están formadas por un gran número de nodos, el costo por nodo es muy importante para justificar el costo total de la red. Si el costo total de la red es mayor al de ofrecer otra solución al problema, entonces no sería viable el utilizar esta tecnología. A causa de esto, el costo por nodo debe de mantenerse bajo.

II.2.4 Limitantes de hardware

Un nodo está compuesto de cuatro componentes básicos (Akyildiz *et al.*, 2002) como se muestra en la Figura 3: unidad de sensado, unidad de procesamiento, unidad de comunicación y fuente de energía. Adicionalmente, en algunos casos se puede contar con otros módulos (no mostrados) dependiendo de la aplicación específica, entre estos módulos se pueden mencionar: módulo de determinación de ubicación (por ejemplo: GPS), módulo de generación de energía (por ejemplo: celdas solares) y módulo de movilización, el cual se puede requerir para mover a los nodos de posición para que éstos realicen las tareas asignadas.

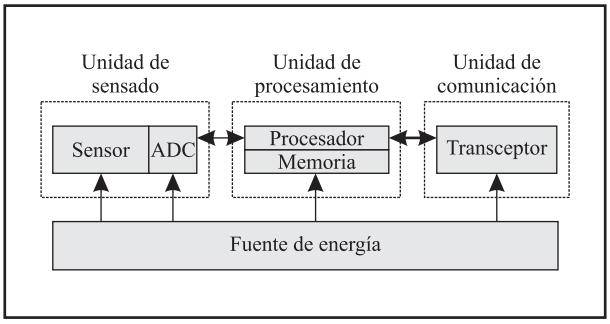

Figura 3. Componentes de un nodo.

La unidad de sensado está integrada por dos componentes: los sensores y los convertidores analógico-digitales (ADC). Las señales analógicas producidas por los sensores con base en el fenómeno observado se convierten a señales digitales por medio

de los ADCs y posteriormente alimentadas a la unidad de procesamiento. La unidad de procesamiento comúnmente posee una pequeña unidad de almacenamiento de datos (memoria); esta unidad lleva a cabo las funciones necesarias para que los nodos colaboren con otros nodos y así desempeñen las labores de sensado asignadas. La unidad de comunicación cuenta con un componente de transmisión-recepción (transceptor), el cual conecta el nodo a la red. Uno de los componentes más importantes de los nodos es la fuente de energía, la cual típicamente está integrada por baterías, las cuales alimentan al resto de los componentes ya mencionados.

Todos estos componentes deben de caber físicamente en un pequeño encapsulado del tamaño de una caja de cerillos aproximadamente (Intanagonwiwat *et al.*, 2000). En algunos casos el tamaño requerido puede ser aún menor a un centímetro cúbico (Pottie y Kaiser, 2000), y lo suficientemente ligero para flotar en el aire (como lo detallan en el proyecto *Smart Dust*, ver Figura 4). Además del tamaño, existen otras limitantes extremas en los nodos. Estos nodos deben consumir muy poca energía (Kahn *et al.*, 1999), operar en redes densamente pobladas, tener un bajo costo de producción, que ninguno de estos nodos sea imprescindible, que sean autónomos, que operen de forma desatendida y que se autoconfiguren de acuerdo a las condiciones que les presente el entorno en el cual están operando.

Actualmente, la plataforma de hardware más utilizada por la comunidad científica es la MICA2 (mostrada en la Figura 5). Estos motes[1] se diseñaron en la Universidad de California en Berkeley y los comercializa la compañía Crossbow[2]. Los motes MICA2 trabajan con RF en las bandas de los 916 y 430 MHz mediante el transceptor CC1000 de Chipcon, cuentan con un microprocesador ATMega128L de 8 MHz de la compañía

[1]Otra forma de llamar a los nodos de sensado que integran WSN.

[2]Sitio web: http://www.xbow.com/, visitado el 24 de octubre del 2009.

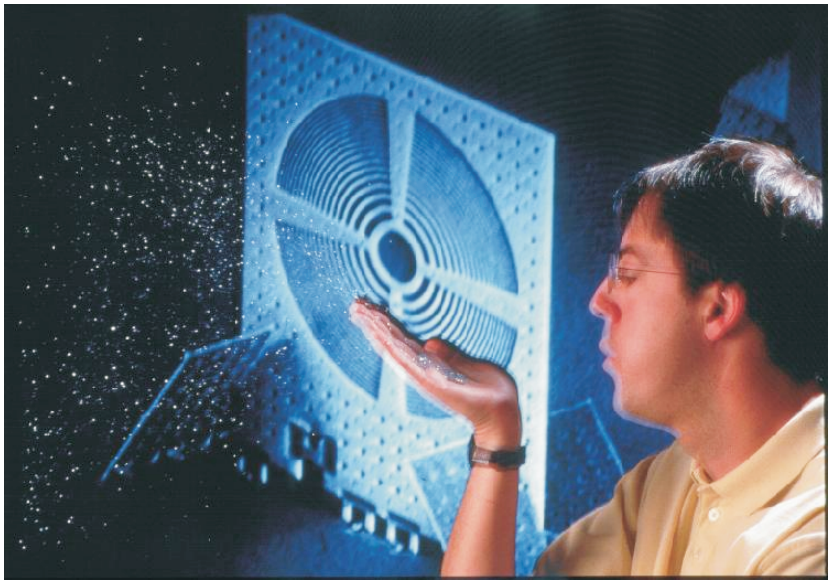

Figura 4. Fotografía de concepto del proyecto Smart Dust.

ATMEL y además cuentan con solo 512 KB de memoria; en cuanto a la fuente de alimentación se refiere, estos motes los alimentan dos baterías comerciales tipo AA. Las dimensiones del mote MICA2 mostrado en la Figura 5 es de 58 mm de largo por 32 mm de ancho por 7 mm de alto.

En lo que se refiere a las capacidades de sensado, se cuenta con varias opciones de módulos que se pueden incrustar a estos motes, la elección del módulo a utilizar se debe hacer con base en los datos que se requieran monitorear, por ejemplo: humedad, temperatura, luminosidad, movimiento, sonido, presión atmosférica, etc. En la Figura 6 se muestra el módulo MTS300CA, el cual cuenta con sensores de temperatura, luminosidad y audio. Este módulo es compatible con el mote MICA2 mostrado en la Figura 5.

II.2.5 Topología

Las redes de sensores están conformadas por un gran número de nodos, los cuales se instalan a unos cuantos metros uno de otro (Intanagonwiwat *et al.*, 2000). La densidad

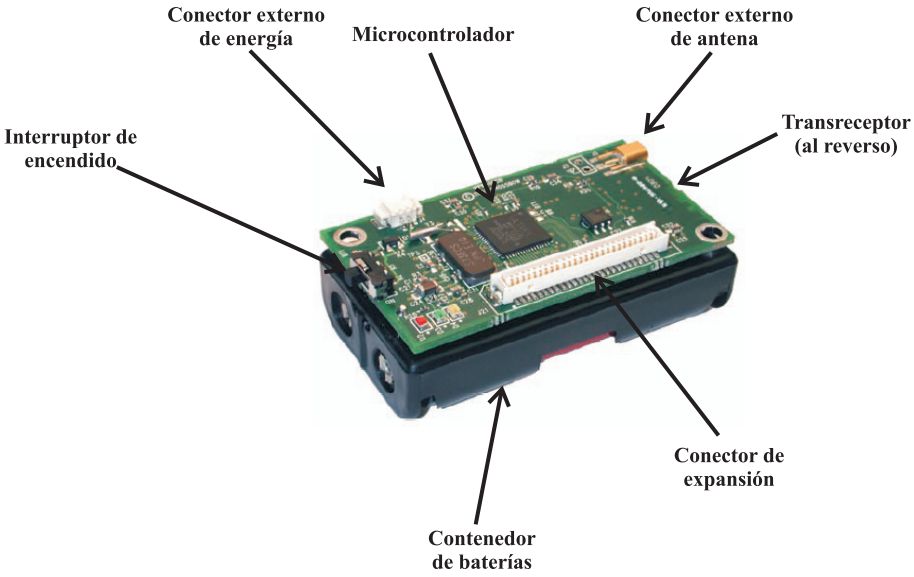

Figura 5. Fotografía de un mote MICA2.

Figura 6. Módulo de sensado MTS300CA.

de los nodos puede llegar a ser de hasta 20 nodos por metro cúbico (Shih *et al.*, 2001). El instalar densamente un alto número de nodos requiere de especial cuidado en el mantenimiento de la topología. Las decisiones relacionadas con el mantenimiento y cambio de topología se presentan a lo largo de tres fases:

- **Fase de Pre-instalación e Instalación**. Los nodos de pueden lanzar en masa (e.g. lanzados desde una aeronave o cohete) o instalar uno por uno en el campo de sensores (e.g. por un humano o por un robot).
- **Fase de post-instalación**. Después de la instalación, se pueden presentar cambios en la topología de acuerdo con cambios en la posición de los nodos, alcance de la transmisión, energía disponible, mal funcionamiento de los nodos, etc.
- **Fase de instalación de nodos adicionales**. Se pueden instalar nodos adicionales para reemplazar nodos que presenten mal funcionamiento o simplemente si existen cambios en la especificación de las tareas a realizar por la red de sensores.

II.2.6 Ambiente

Los nodos se instalan masivamente muy cerca o inclusive dentro del fenómeno (por ejemplo dentro del cráter de un volcán) a observar. Por tal motivo, comúnmente operan en forma desatendida en áreas geográficas remotas. Por ejemplo: dentro de maquinaria, en el fondo del océano, en un campo contaminado biológica o químicamente, en un campo de batalla, en una casa o en un edificio.

II.2.7 Medio físico de transmisión

En una red de sensores, los nodos se comunican utilizando enlaces a través del medio inalámbrico. Estos enlaces se pueden establecer por medio de radiofrecuencia, infrarrojo

u ópticos.

La mayoría del hardware actual para redes de sensores se basa en el uso de radiofrecuencia (RF). El nodo μAMPS descrito por Shih *et al.* (2001) se basa en tecnología compatible con Bluetooth que opera con un transceptor a 2.4 GHz. El dispositivo descrito por Woo y Culler (2001) utiliza un canal de RF que opera en los 916 MHz. La arquitectura WINS (*Wireless Integrated Network Sensors*) (Pottie y Kaiser, 2000) también utiliza enlaces de RF para comunicación.

Otra forma posible de comunicación en redes de sensores es mediante luz infrarroja. La comunicación infrarroja está libre de licenciamiento y además es resistente a interferencia de dispositivos eléctricos. Además, los transceptores infrarrojos son muy económicos y fáciles de construir. Otra opción interesante de comunicación es la óptica; esta opción la utilizan los motes Smart Dust (Kahn *et al.*, 1999). El gran inconveniente de estas últimas dos opciones (infrarroja y óptica) es que ambas requieren de línea de vista para su operación.

II.2.8 Consumo de energía

Los nodos debido a su tamaño reducido, solo pueden tener una pequeña fuente de alimentación. En la mayoría de las aplicaciones, la recarga de energía no es factible. Por tal motivo, la vida de los nodos está íntimamente relacionada con la vida de la batería. En una red de sensores multisaltos, los nodos desempeñan dos papeles: generar datos sensados y rutear tráfico generado por otros nodos. El mal funcionamiento de los nodos puede causar cambios en la topología y con eso la necesidad de reorganizar el ruteo del tráfico. Por tal motivo, es importante el ahorro de energía. El desarrollar trabajo de investigación relacionado con protocolos y algoritmos eficientes en el consumo de energía

es un reto que ha estado afrontando la comunidad científica (van Dam y Langendoen, 2003; Klues *et al.*, 2007).

La tarea principal de un nodo es la de detectar eventos, procesar datos y transmitirlos. De tal manera, que el consumo de energía se puede dividir en tres partes: sensado, comunicación y procesamiento de datos.

II.3 Pila de protocolos

En (Akyildiz *et al.*, 2002) se presenta una pila de protocolos de referencia para redes inalámbricas, ésta consta de cinco capas: la capa de aplicación, de transporte, de red, de acceso al medio y la capa física, de forma similar a la pila de protocolos de las redes tradicionales; además se incluyen tres planos, generando así una pila tridimensional: los planos de administración de energía, de movilidad y de tareas. En la Figura 7 se muestra la pila de protocolos.

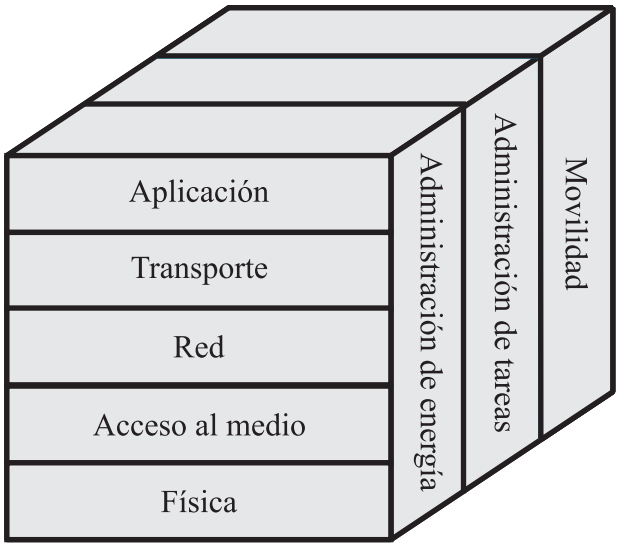

Figura 7. Pila de protocolos de las redes inalámbricas de sensores.

Como se puede observar en la Figura 7, los tres planos se extienden por todas las

capas de la pila de protocolos. El plano de energía define cómo los nodos administran su energía mediante sus protocolos en las diversas capas, por ejemplo: un nodo puede apagar su receptor inmediatamente después de recibir un mensaje de alguno de sus vecinos, este caso cae en la capa de acceso al medio. Otro caso de administración de energía es por ejemplo, que cuando un nodo llegue a cierto nivel de energía deje de participar en tareas de ruteo y se concentre únicamente en su función principal que es la de sensado del fenómeno determinado; se puede observar que esto cae dentro de la capa de red. El plano de tareas se encarga de balancear y calendarizar las tareas de sensado para una región específica dada. No todos los sensores pertenecientes a una región requieren tomar lecturas al mismo tiempo. De tal manera, que únicamente tendrían que efectuar la tarea de sensado los nodos que cuenten con un alto nivel de energía disponible. Los planos de administración son necesarios para que los nodos operen de manera eficiente en cuanto al consumo de recursos (particularmente energía). El plano de movilidad detecta y registra el movimiento de los nodos de sensado, de tal forma que siempre se mantenga una ruta de regreso hacia el usuario, además, este plano se encarga de que los nodos estén al tanto de quíenes son sus vecinos. Al saber quíenes son sus vecinos, los nodos pueden balancear el uso de energía.

La pila de protocolos presentada en la Figura 7, al igual que la pila de protocolos tradicional, sugiere una separación de las funciones entre los protocolos de las diferentes capas. Sin embargo, en el caso de las redes inalámbricas de sensores, esto desfavorece la parte de optimización de recursos ya que el bajo acoplamiento de los protocolos implica que realizan sus funciones sin considerar detalles de la operación de protocolos de capas inferiores. De tal manera que, existen diversos esfuerzos de investigación basados en el diseño de conjuntos de protocolos que presentan alto acoplamiento entre ellos, esto con el fin de considerar los detalles de la funcionalidad de cada uno de ellos eliminando

operaciones redundantes; a este paradigma se le conoce en la literatura como diseño *cross-layer*. Por lo anterior, se considera altamente probable que la mejor solución para el diseño de protocolos es el paradigma *cross-layer* ya que va acorde con las extremas limitantes de recursos que se presentan en las redes inalámbricas de sensores.

En este capítulo se abordaron algunas de las generalidades más significativas acerca de las WSN, entre ellas, su descripción general, sus características y limitaciones, y su pila de protocolos; citando el trabajo previo relacionado con los contenidos ya descritos. Por otra parte, en el siguiente capítulo se describen los efectos adversos que puede presentarse en aplicaciones de WSN que manejan información redundante, también se presentan los trabajos previos más relevantes que abordan el problema del manejo de información redundante en WSN.

Capítulo III

Redundancia de Datos en WSN

En este capítulo se presente el uso general de redundancia en el área de ingeniería, se muestra que la redundancia por un lado mejora la confiabilidad de los sistemas en general. Por este motivo, en el ámbito de redes inalámbricas de sensores también se emplea la redundancia de nodos. Sin embargo, al hacer esto, se muestra que ciertas aplicaciones para WSN presentan información redundante que puede afectar en forma negativa el desempeño de las mismas. Esto se acentúa particularmente en estas plataformas, donde como ya se ha mencionado anteriormente, algunos recursos como ancho de banda, poder de procesamiento, almacenamiento y fuente de energía son extremadamente escasos. Así también, se presentan algunos trabajos previos con resultados preliminares que abordan el problema del manejo de datos redundantes en redes inalámbricas de sensores.

III.1 Generalidades sobre Redundancia

En ingeniería, la redundancia es la duplicidad de componentes críticos de un sistema con la intención de incrementar la confiabilidad del mismo.

En muchos sistemas donde la seguridad es crítica, como en el caso de los sistemas hidráulicos en los aviones, algunas partes de los sistemas de control pueden estar inclusive triplicadas. Un error en uno de los componentes hace que el de respaldo entre en acción. En un sistema de triple redundancia, el sistema cuenta con tres subcom-

ponentes, los tres tendrían que fallar para que el sistema en su totalidad fallase. Y como en este tipo de aplicaciones la probabilidad de que falle uno es baja, y los subcomponentes se espera que fallen en forma independiente, la probabilidad de que los tres fallen simultáneamente es mucho más pequeña. La redundancia también se conoce como "sistemas de votación mayoritaria" (Flavin, 1991) o "lógica de votación". Los sistemas de votación mayoritaria se utilizan para proteger sistemas críticos. Dichos sistemas se pueden encontrar en diversos sectores: químico, energético, nuclear, aeroespacial y muchos otros más. Aquí múltiples sensores monitorean un proceso crítico, y las lecturas de dos sensores por ejemplo se toman para monitorear el sistema; y si un nodo fallase, el proceso crítico se seguiría monitoreando.

Con cada componente duplicado que se vaya agregando al sistema se decrementa la probabilidad de falla en el sistema, o sea que:

$$P = \prod_{i=1}^{n} p_i \tag{2}$$

donde:

- n: número de componentes.
- p_i: probabilidad de que el componente i falle.
- P: probabilidad de que todos los componentes fallen simultáneamente (falla del sistema).

La Ecuación 2 supone independencia en los eventos de falla en los componentes. Esto significa que las probabilidad de que falle un componente B dado que ha fallado un componente A es la misma que la probabilidad de que falle B aun cuando A no haya fallado. Hay situaciones donde esto no es posible, por ejemplo, en el caso donde

dos fuentes de poder redundantes utilizan el mismo conector y este último falla. En la ecuación también se supone que solo se requiere de un componente para que el sistema funcione. Si se requiere de m de un total de n componentes para que el sistema continúe operando, la probabilidad de falla es $1 - ((1-p)^{(m-n)}({}_nC_m))$, suponiendo que todos los componentes tienen la misma probabilidad p de falla.

Dentro de la redundancia en ingeniería, se identifican cuatro principales tipos de redundancia:

1. Redundancia de hardware, tal como la DMR (*Redundancia Modular Dual*) y la TMR (*Redundancia Modular Triple*)

2. Redundancia de información, tal como la utilizada en los mecanismos de detección y corrección de errores.

3. Redundancia en tiempo, como ciertos mecanismos de detección de fallas.

4. Redundancia de software, como en el caso del enfoque NVP (*Programación de N-versiones*).

En la actualidad, la redundancia de componentes se puede encontrar hasta en productos de consumo: baterías de respaldo para sistemas de cómputo, fuentes de energía redundantes (ver Figura 8); en sistemas de información: bases de datos replicadas, arreglos redundantes de discos duros (arreglos de tipo RAID); y hasta en la naturaleza: genes redundantes que aportan la misma información en ciertos organismos (Kafri *et al.*, 2009).

En el caso de las WSN, los nodos de sensado son muy susceptibles a fallas. Por tal motivo, es común que se haga uso de la redundancia para mejorar la confiabilidad general de la red. Como se menciona en el resto capítulo, el tener nodos redundantes

Figura 8. Fuente de poder redundante.

por un lado mejora la confiabilidad, pero por otro genera información redundante, lo cual no es bueno, debido a las limitaciones de recursos que ya se han mencionado.

III.2 Aplicaciones orientadas a datos y orientadas a eventos

Las aplicaciones de redes de sensores pueden pertenecer a uno o a los dos siguientes grupos (Bulusu y Jha, 2005):

- *Orientadas a datos.* Aquí las aplicaciones recolectan y analizan datos del ambiente, y dependiendo de la redundancia que se presenta, el ruido y las propiedades de los sensores mismos, los datos tienen cierto valor para las aplicaciones.

- *Orientadas a eventos.* Estas aplicaciones pueden estar interesadas en eventos es-

pecíficos, y el objetivo principal de la red de sensores es el de detectar e identificar tales eventos.

Frecuentemente para tales aplicaciones orientadas a datos y/o eventos, ciertas tareas básicas tales como el ruteo, almacenamiento y diseminación se deben modificar para tomar en cuenta el contenido y la estructura de los datos en lugar de tratarlos como secuencias aleatorias de bits. Además, la noción de eficiencia es diferente para los dos enfoques descritos, donde el valor de los datos y no la cantidad de datos transferidos a la red es importante. Debido a que los nodos alimentados por baterías cuentan con un tiempo de vida muy limitado durante el cual proveen de datos a la aplicación, un reto para el diseño de redes de sensores es el de maximizar el tiempo de vida de la red (lo cual implica que los sensores y la red sean eficientes en el consumo de energía), mientras satisfacen requerimientos de la aplicación tales como calidad de los datos (medida en términos de tolerancia de errores, relación señal a ruido, resolución o alguna otra métrica especifica de la applicación) y latencia.

Normalmente se contraponen entre sí ciertos parámetros, por ejemplo la eficiencia energética y la latencia (cuando los nodos de sensado operan a un bajo ciclo de trabajo para ahorrar energía, se retrasa la notificación de eventos). Otro ejemplo es cuando una menor cantidad de nodos envían datos para ahorrar energía se reduce la disponibilidad de los datos. Para aplicaciones orientadas a datos y a eventos, las necesidades de la aplicación (las cuales pueden variar conforme pasa el tiempo en base al estado del fenómeno que se esté monitoreando) deben dictar de qué lado se debe equilibrar la balanza en cuanto a eficiencia energética, calidad de los datos y latencia se refiere. Muchas veces, este balanceo se puede mapear directamente a cómo la red y los mismos nodos de sensado mismos se configuran, administran y operan.

III.3 Redundancia de datos en WSN

De entre los dos grandes grupos de aplicaciones de WSN mencionados en la sección anterior, ciertas aplicaciones dentro de las catalogadas como orientadas a datos, operan reportando en forma períodica el estado del ambiente que están monitoreando. Este reporte, generalmente se da a períodos de tiempo constante, sin importar si ocurre o no un cambio en el fenómeno que se encuentran monitoreando. Desde la perspectiva de la estación base, el recibir un reporte de un conjunto de nodos, se puede utilizar como un aviso de que tal conjunto de nodos aún permanece activo. Sin embargo, en la mayoría de los casos, dichos reportes no son más que información redundante debido a que si no se ha presentado una circunstancia extraordinaria, todos los nodos que forman parte del conjunto mencionado, reportan básicamente el mismo estado de las variables ambientales. El hecho de estar reportando datos redundantes, puede potencialmente llevar al derroche de recursos en una plataforma ya de por sí limitada. Los recursos que se verían afectados al caer en la transmisión de datos redundantes son los siguientes:

- *Fuente de energía.* Al transmitir en forma innecesaria datos redundantes, lo primero que sufre es la fuente de alimentación de los nodos de sensado, ya que la interfaz de comunicaciones es el módulo que mayor cantidad de energía consume. Sin embargo, se vislumbra que en un futuro no muy lejano, esto no sea el recurso más importante ya que recientemente se han publicado trabajos relacionados a hacer que los nodos adquieran su propia energía del ambiente (Roundy *et al.*, 2004; Kansal *et al.*, 2004) (conocido como *energy harvesting* en inglés).

- *Almacenamiento.* El estar sensando datos del medio constantemente puede reducir considerablemente la capacidad de almacenamiento de datos de los nodos, esto debido a que si el nodo de sensado en sí no tiene la capacidad de detectar si

los datos que va sensando son redundantes, simplemente los almacena en memoria para su posterior transmisión.

- *Procesamiento.* Aunque quizás en menor escala, la capacidad de procesamiento de datos también se ve afectada cuando no hay un manejo adecuado de la redundancia en WSN, esto debido al hecho de que conforme aumenta la cantidad de datos a transmitir, la carga de la unidad de procesamiento también aumenta, ya que ésta es la encargada de manipular los datos (mover datos de localidades de memoria, transferirlos a los buffers de comunicación, etc.).

- *Ancho de banda.* En este tipo de redes, otro de los recursos que más sufre cuando se presentan transmisiones de datos innecesarias es la capacidad de comunicación de la red. Al decir esto, se puede establecer principalmente que el introducir tráfico adicional a la red, merma considerablemente el ancho de banda de la WSN, introduce retardos, aumenta la contención por el canal de comunicaciones ocasionando pérdida de paquetes. Todo esto, desde la perspectiva de la aplicación puede tener un impacto negativo en la confiabilidad del sistema en general.

Por lo que se acaba de mencionar, el manejo de la redundancia de datos es importante en todas las aplicaciones de WSN. Sin embargo, en aplicaciones de red que manejan altas tasas de datos tales como señales de audio y/o vídeo, esto cobra aún mayor importancia. El alto volumen de datos generado por sensores de ese tipo de aplicaciones hacen imperativo el extraer y transmitir solo lo que es absolutamente indispensable, inclusive, a medida de lo posible, transmitir únicamente descripciones de alto nivel de lo que se está sensando y no las señales puras captadas por los nodos. Por otra parte, como se menciona posteriormente, los datos sensados en nodos cercanos están altamente correlacionados con frecuencia; por tal motivo, tal redundancia se debe

detectar y remover.

El manejo de redundancia en redes inalámbricas de sensores se ha abordado con anterioridad. Sin embargo, no se puede establecer una solución única para la detección y eliminación de datos redundantes, ya que como sucede en el caso del diseño de protocolos de comunicación para WSN, el diseño de la técnica está íntimamente ligado a la naturaleza de la aplicación; si el propósito de la aplicación cambia, muy probablemente tengan que cambiarse las características funcionales de las técnicas de bajo nivel que fueron diseñadas para tal aplicación.

Enseguida se presentan algunas técnicas mencionadas en la literatura con la finalidad de eliminar (o simplemente reducir) las transmisiones de datos redundantes en redes inalámbricas de sensores.

III.4 Administración de tareas con base en información sensada

La idea detrás de la administración de tareas con base en información sensada (*Information-Based Sensor Tasking* en inglés), es seleccionar los nodos que participarán en una tarea específica con base en el contenido de la información así como en las limitantes de consumo de recursos, latencia y otros costos. Al utilizar la información como un factor de decisión, los nodos de la red pueden explotar el contenido de la información que ya ha sido recibida con anterioridad para optimizar la utilidad de acciones posteriores de sensado y comunicaciones, y por consiguiente, administrar eficientemente los escasos recursos de comunicación y procesamiento. Por ejemplo, *IDSQ* (Chu *et al.*, 2002; Zhao *et al.*, 2002) establece el problema de administración de tareas como un problema de

optimización que maximiza la cantidad de información proveniente de los nodos de sensado minimizando el uso de recursos de comunicación.

Indudablemente, la parte central de este enfoque es la selección de nodos, es decir, cómo seleccionar el nodo que pueda generar la información más útil, incurriendo en el menor costo de recursos posible.

III.5 Diseminación de datos altamente correlacionados en WSN utilizando un enfoque bio-inspirado

En este trabajo se adaptó un algoritmo bio-inspirado llamado "algoritmo de infección" (Olague *et al.*, 2004) para la diseminación de datos del nodo de sensado a la estación base. Aquí se presentan resultados de experimentos que se llevaron a cabo con datos reales, recolectados con una aplicación de monitoreo de un invernadero de tomates, los experimentos y las simulaciones validan la eficiencia energética de dicha propuesta (Palafox y García-Macías, 2006). En este trabajo se presenta la integración de algoritmos bio-inspirados en redes inalámbricas de sensores, este enfoque representa una alternativa interesante que puede dar como resultado protocolos de comunicación eficientes debido a que en su mayoría, estos algoritmos trabajan con información local, al igual que las WSN. En el Capítulo IV se presentan más detalles de la propuesta de la utilización del algoritmo de infección, así como resultados preliminares generados a partir de una serie de experimentos y simulaciones.

III.6 Agregación segura de datos basada en diferencias

Çam *et al.* (2006) presentan un esquema de agregación de datos segura diferencial para WSN (SDDA). Este esquema se basa en transmitir solo las diferencias en información con el fin de minimizar las operaciones de transmisión.

Los autores mencionan que la agregación de datos en WSN es esencial debido al hecho de que en algunos casos la misma área la cubren más de un nodo. En los algoritmos de agregación de datos convencionales, los nodos de sensado transmiten los datos puros al agregador. En este caso, el protocolo propuesto aquí transmite los datos diferenciales en lugar de transmitir todos los datos actuales, al decir datos diferenciales, se refiere a la diferencia de los datos actuales con un dato de referencia. Suponiendo que los valores de las lecturas de los sensores se mapean a números, el valor de referencia que usan en ese trabajo es el promedio de todas las lecturas en la sesión de transmisión más reciente. Cada nodo calcula su referencia y la envía al nodo líder. Por ejemplo, considerese que la lectura actual de un nodo dado es de 102° F. Si la temperatura de referencia es de 100° F, el nodo podría enviar sólo la diferencia (2° F) como su lectura actual. Por lo tanto, los autores consideran que la agregación diferencial cuenta con gran potencial de poder reducir la cantidad de datos a ser transmitidos de los nodos de sensado hacia el nodo líder. La motivación principal detrás de la agregación diferencial es que los cambios significativos en las lecturas solo ocurren ante la presencia de eventos críticos (por ejemplo: un incendio en un sistema de monitoreo forestal) en el ambiente. En general, en redes de sensores, estos eventos críticos ocurren con mucha menos frecuencia que los eventos ordinarios. Para aprovechar las ventajas de la agregación diferencial en cada transmisión del nodo de sensado al nodo líder, el

protocolo propuesto hace uso de códigos de patrones así como los datos actuales como se explica enseguida.

También el nodo líder se beneficia de la agregación de datos, ya que éste recibe y procesa una menor cantidad de datos. La eficiencia de la técnica propuesta aumenta, conforme aumenta el volumen de los datos, ya que el valor de referencia se transmite una sola vez. Para poder utilizar agregación diferencial, el protocolo SDDA determina los valores de referencia para los códigos de patrones y para el valor actual tomando en cuenta lecturas pasadas. Después, los nodos líderes y la estación base almacenan estos valores de referencia para poder recuperar los códigos de patrones de las lecturas actuales y sus datos correspondientes, con solo recibir los patrones de código y los datos diferenciales. SDDA primero genera códigos de patrones al identificar los rasgos más importantes de las lecturas de los nodos y luego determina el código del patrón diferencial al obtener la diferencia entre el código del patrón actual y el código del patrón de referencia. Posteriormente, el código del patrón diferencial se transmite al nodo líder. Cuando el nodo líder realiza la petición de datos actuales, el nodo de sensado le transmite los datos diferenciales actuales.

III.6.1 Algoritmo de agregación de datos diferencial

El algoritmo de agregación de datos diferencial se describe en las siguientes tres fases:

- *Fase 1.* Se envían valores de referencia para los códigos de patrones hacia el nodo líder, se envían valores de referencia para los datos actuales hacia la estación base por medio del nodo líder (ver Figura 9) al inicio de cada sesión de transmisión. Después de sensar datos nuevos, el nodo de sensado genera códigos de patrones actuales al encontrar las principales características de los datos sensados. Poste-

riormente, el nodo de sensado le transmite el código del patrón diferencial al nodo líder.

- *Fase 2.* Después de recibir todos los códigos de patrones diferenciales, el nodo líder recupera todos los códigos de patrones actuales al sumar los códigos de patrones de referencia. Posteriormente, el nodo líder determina códigos de patrones distintos y solicita sólo a un nodo el transmitir los datos para cada código de patrón.

- *Fase 3.* Cada uno de los nodos de sensado que recibieron una petición de datos del nodo líder genera los datos diferenciales al encontrar la diferencia entre los datos sensados y sus datos de referencia. Posteriormente, los datos diferenciales se encriptan y se envían al nodo líder, el cual a su vez los envía a la estación base. Los datos diferenciales se utilizan para obtener los datos actuales en la estación base utilizando los datos de referencia del nodo de sensado.

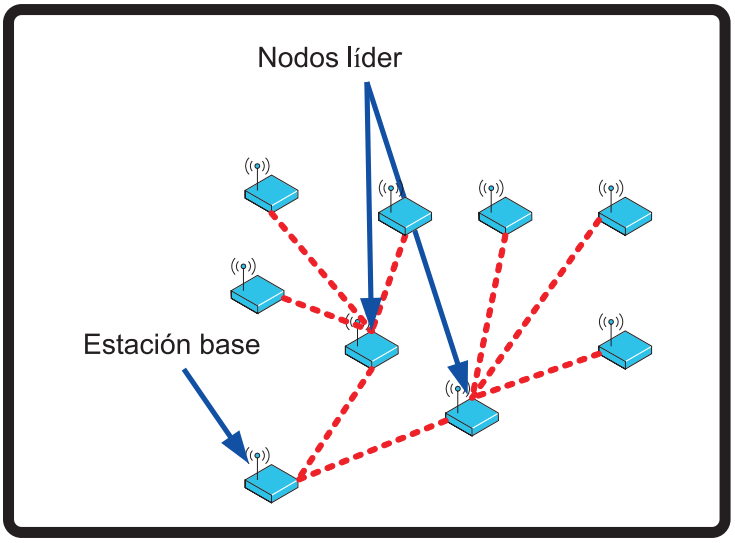

Figura 9. Red Inalámbrica de Sensores Jerárquica.

Como se pudo ver en este capítulo, después de realizar una revisión exhaustiva de la literatura, realmente se encontró poco trabajo que aprovecha la existencia de

redundancia de datos en redes inalámbricas de sensores, debido a esto, se identificó una oportunidad de contribución original que se aborda en el siguiente capítulo. En el siguiente capítulo, se presenta una propuesta de integración de un algoritmo bio-inspirado para WSN, que si bien, no es la parte principal de este trabajo de tesis, si resultó una contribución adicional dentro del desarrollo del mismo. La utilización de técnicas no convencionales como las bio-inspiradas representan una oportunidad de investigación importante que presenta resultados atractivos.

Capítulo IV

Un Enfoque Bioinspirado para la Diseminación de Datos en WSN

En los últimos años, los algoritmos bio-inspirados se han presentado como una alternativa para el diseño de varios aspectos de redes ad-hoc y redes inalámbricas de sensores. En este capítulo, se presenta la propuesta de adaptación de un algoritmo bio-inspirado existente, al cual se le llama "algoritmo de infección". La propuesta se enfoca en la diseminación de datos eficiente en el consumo de energía que va del campo de sensado al nodo sumidero. Además, se presentan una serie de experimentos y simulaciones con datos recolectados en una implementación real de una red inalámbrica de sensores en una aplicación de monitoreo agrícola dentro de un invernadero de tomates, los resultados presentados en este capítulo validan la eficiencia de la propuesta.

IV.1 Enfoque Bioinspirado en Redes Inalámbricas de Sensores

Dadas las extremas limitaciones que presentan las WSN, no es factible utilizar las mismas técnicas y algoritmos que se utilizan en redes tradicionales ya que éstas no eficientizan el uso de los recursos disponibles. Por lo anterior, se han explorado nuevos enfoques, entre ellos el enfoque bio-inspirado (Britton *et al.*, 2005).

Por medio de un estudio empírico conducido por Ganesan *et al.* (Ganesan *et al.*, 2002) se introdujo el concepto de algoritmos epidémicos con el fin de describir el comportamiento de protocolos de red que permiten la rápida diseminación de datos por medio del uso de interacciones locales. En este trabajo, se construyó una red de sensores de 169 nodos (una malla de 13x13), posteriormente se desarrollaron una serie de experimentos utilizando un simple algoritmo epidémico para la diseminación de datos. La parte interesante de este trabajo es el hecho de que un algoritmo tan simple puede presentar un comportamiento tan complejo, según lo muestran los autores en la sección de resultados de su reporte.

También se ha propuesto el uso de autómatas biológicos para definir el comportamiento y las interacciones entre los nodos en una red inalámbrica de sensores (Britton *et al.*, 2005). Esto se llevó a cabo mediante el diseño e implementación de un kOS (Sistema Operativo de kilobit por sus siglas en inglés), el cual es una versión ligera de un sistema operativo que ejecutan los nodos de una red de sensores. Los autores presentan también los resultados obtenidos del proyecto SEACOAS[1], en el cual se utilizan boyas que contienen nodos de sensado con módulos de RF (para comunicación de datos) con el fin de monitorear el movimiento que presenta la superficie oceánica.

También se han introducido ciclos de retroalimentación con la finalidad de proveer mecanismos de autoconfiguración en los nodos de sensado (Dressler *et al.*, 2005). En este trabajo, se utiliza como modelo el proceso de auto-regulación de la presión arterial del cuerpo humano, el cual, mencionan los autores tiene una estructura de retroalimentación en ciclo cerrado. Este tipo de comportamiento se utiliza para configurar automáticamente una red inalámbrica de sensores asistida por robots, tal como la

[1]Sitio web del proyecto: http://www.cs.kent.ac.uk/projects/secoas/, visitado el 13 de marzo del 2009.

plataforma ROSES (Robot Assisted Sensor Network), la cual se describe brevemente en el artículo (Dressler *et al.*, 2005).

En el trabajo presentado en (Werner-Allen *et al.*, 2005), los autores utilizan el proceso de sincronización espontánea de las luciérnagas y proponen el Algoritmo de Sincronización de Luciérnagas (RFA, *Reachback Firefly Algorithm*), el cual se utiliza para sincronizar redes inalámbricas de sensores en el tiempo. Adicionalmente, este algoritmo considera los efectos reales introducidos por las unidades de RF tales como la pérdida de paquetes y la latencia. Como queda evidenciado por lo que se menciona en los párrafos anteriores, el uso de algoritmos basados en sistemas biológicos han generado resultados por demás interesantes. Por tal motivo, este tipo de algoritmos constituyen una herramienta que representa un gran nicho de investigación en el diseño de tecnologías para redes inalámbricas de sensores.

IV.2 El algoritmo de infección

El "*algoritmo de infección*", presentado originalmente por (Olague *et al.*, 2004), se basa en el concepto de la epidemiología natural y fue utilizado para la búsqueda de puntos de correspondencia en imágenes stereo, logrando reducir el número de operaciones de procesamiento requeridas en comparación al método tradicional de búsqueda exhaustiva.

Se utilizaron reglas de transición para la búsqueda de correspondencias, en forma similar a cómo se hace en autómatas celulares. Las entradas de las reglas dependen del estado actual de los vecinos de alrededor (pixeles). La vecindad considerada en este trabajo estaba conformada por 25 vecinos (9 vecinos cercanos y 16 externos), estos 25 vecinos se encontraban dentro de una ventana de 7×7 que estaba centrada al punto

de interés.

El proceso de infección en este caso evolucionaba sobre la imagen de acuerdo al conjunto de reglas ya mencionadas que se encargan de cambiar el estado actual del píxel dependiendo del estado de los vecinos que lo rodean. Se definen cuatro estados para este algoritmo:

- Individuos sanos (No expuestos). Nada se ha decidido aún para este píxel.
- Individuos enfermos (Expuestos). El píxel ha sido procesado siguiendo las reglas.
- Individuos infectados (Propuestos). El valor del píxel se infiere basándose en el estado de sus vecinos. Cierta información dudosa proveniente de varios vecinos evita que se le asigne un estatus en este momento.
- Individuos inmunes (Automáticamente asignados). Toda la información del vecino es consistente y el valor inferido ya se ha asignado.

Concretamente, el algoritmo se define de la siguiente manera:

1. Todos los pixeles de la imagen se inicializan en el estado *No expuesto*.
2. Se extraen los pixeles de máximo interés de la imagen y se les asigna el estado *Expuesto*.
3. Se aplican las reglas de transición a cada píxel de la imagen, excepto a aquellos cuyo estado es *Automáticamente asignado* o *Expuesto*.
4. Mientras existan pixeles que no estén en el estado *Automáticamente asignado* o *Expuesto*, ir al paso 3.

El objetivo principal de este algoritmo es el de encontrar el número máximo de puntos de correspondencia de acuerdo a las reglas definidas. Los autores mencionan que las reglas se definen caso por caso, pero el criterio para definir estas reglas no se especifica en ese trabajo.

Un aspecto interesante de esta propuesta es el hecho de que en comparación con el método de la búsqueda exhaustiva, el algoritmo propuesto presenta ahorros de hasta el 50% en el número de operaciones (en algunos casos hasta el 99%). En la propuesta que se presenta en este capítulo se considera que las ideas generales introducidas por este algoritmo pueden resultar útiles en redes inalámbricas de sensores. Por ejemplo, este algoritmo actúa basándose en información local, estimando valores en la vecindad de un píxel; esto se puede trasladar a ahorros importantes en procesamiento, lo cual en WSN puede resultar también en ahorro en el consumo de energía. En aplicaciones donde se requiere de WSN densamente pobladas por nodos que sensen ciertos parámetros ambientales tales como temperatura o humedad, es altamente probable que los datos sensados por un nodo sean muy parecidos a los sensados por un vecino inmediato. Por lo tanto, resulta muy atractivo el utilizar el algoritmo de infección para leer solo un pequeño número de nodos y estimar los valores de sus vecinos, lo cual podría resultar en importantes ahorros en el consumo de energía.

IV.3 Diseminación de datos eficiente

La diseminación de datos del campo de sensado al nodo sumidero es uno de los problemas principales dentro de la pila de protocolos de las WSN (Akyildiz *et al.*, 2002). Ya existe trabajo previo relacionado con este problema, entre ellos se encuentra Cougar (Demers *et al.*, 2003; Yao y Gehrke, 2002, 2003) y TinyDB(Madden y Hellerstein, 2002;

Madden *et al.*, 2002a,b), estos trabajos presentan el uso de un lenguaje declarativo muy similar a SQL para realizar búsquedas de datos en la red de sensores. Sin embargo, en estos trabajos los autores no entran en detalles en cómo las búsquedas se procesan en la red.

También ha habido trabajo relacionado con agregación de datos en WSN y la utilización de técnicas de estimación para reducir el consumo de energía. Particularmente, en (Boulis *et al.*, 2003) los autores presentan un algoritmo distribuido que utiliza funciones de correlación para estimar el valor agregado y reducir el consumo de energía, pero en este caso, los autores solo consideran funciones de agregación escalares tales como *max*, *min* y no se puede aplicar a otro tipo de funciones como promedio (*avg*) o cuenta (*count*).

En este capítulo se presenta un método alternativo inspirado por el algoritmo de infección que se menciona anteriormente; este método representa una alternativa interesante para la diseminación de datos guiada por el ahorro en el consumo de energía, el cual es el principal factor de diseño en las diferentes capas de la pila de protocolos de las redes inalámbricas de sensores (Akyildiz *et al.*, 2002).

El enfoque original del algoritmo de infección es el de reducir el número de operaciones de procesamiento requeridas. Sin embargo, en el caso de las redes de sensores, la prioridad principal es la eficiencia en el consumo de energía; por tal motivo, el principal objetivo de la propuesta es el de reducir la cantidad de operaciones que involucran a la unidad de RF y no necesariamente las operaciones de procesamiento. Esto debido a que se ha observado una tendencia en incrementar el poder de procesamiento en las redes de sensores (Hill *et al.*, 2004).

Una forma de reducir el número de operaciones de comunicación es que al momento de hacer la petición de datos a un grupo de nodos en una red, no pasar la búsqueda

a todos los nodos que pertenecen a dicho grupo. Por tal motivo, la propuesta que se presenta consiste en seleccionar un subconjunto de nodos que pertenecen a la red jerárquica (mostrada en la Figura 10), donde cada uno de esos nodos tiene una conexión directa al nodo líder del grupo. En el subconjunto de nodos seleccionados se hace una petición explícita de información mientras que en el resto de los nodos se emplearía una estimación basada en el algoritmo de infección.

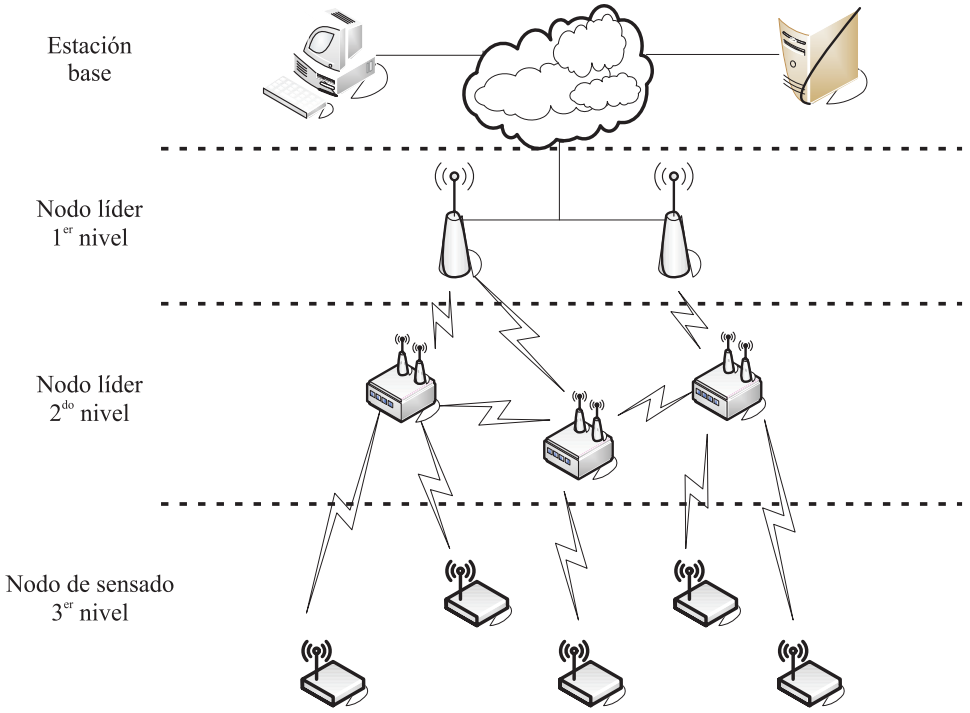

Figura 10. Ejemplo de una red inalámbrica de sensores estructurada jerárquicamente.

La parte más importante del algoritmo de infección para WSN que se presenta en este capítulo es cómo se lleva a cabo el proceso de infección; este proceso se basa en la información que proviene de los vecinos inmediatos de los nodos de sensado. Para estimar los valores de un nodo desconocido se utiliza la correlación con sus vecinos, esta correlación la calcula y almacena el nodo líder utilizando valores de lecturas anteriores. Específicamente, en la propuesta se calcula una matriz de correlación para cada uno

de los ocho nodos vecinos que rodean al nodo de interés (se podrían considerar más de ocho vecinos para una mejor estimación a cambio de un costo computacional más alto y de mayor espacio requerido). Las matrices de correlaciones se deben almacenar en el nodo líder, y este mismo nodo debe calcular sus valores iniciales al efectuar una lectura inicial en todos los nodos de la red que pertenecen al grupo, y posteriormente, este nodo líder también debe actualizar las matrices almacenadas utilizando los valores de lecturas posteriores provenientes de los nodos contiguos pertenecientes al mismo grupo de la WSN.

Las matrices de correlación N, S, E, W, A, B, C, D corresponden a los vecinos al norte, sur, este, oeste, noroeste, noreste, suroeste, sureste respectivamente. Estos nodos tienen una estructura de malla de m filas por n columnas. Cada nodo debe tener un enlace directo a su nodo líder. Por otra parte, $r_{(i,j)}$ representa la lectura en el nodo ubicado en la fila i y en la columna j.

La Ecuación 3 es la matriz N, nótese que la primera fila es un vector de ceros, esto se debe a que los nodos de sensado que corresponden a la primer fila de la malla no cuentan con vecinos al norte.

$$N = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{r_{(2,1)}}{r_{(1,1)}} & \frac{r_{(2,2)}}{r_{(1,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(1,n)}} \\ \frac{r_{(3,1)}}{r_{(2,1)}} & \frac{r_{(3,2)}}{r_{(2,2)}} & \cdots & \frac{r_{(3,n)}}{r_{(2,n)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{r_{(m,1)}}{r_{(m-1,1)}} & \frac{r_{(m,2)}}{r_{(m-1,2)}} & \cdots & \frac{r_{(m,n)}}{r_{(m-1,n)}} \end{bmatrix} \tag{3}$$

Similarmente, en la Ecuación 4 se tiene la matriz S, en la cual la última fila de la malla no tiene vecinos al sur, por lo tanto la matriz de correlaciones tiene ceros en la última fila:

$$S = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(2,1)}} & \frac{r_{(1,2)}}{r_{(2,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(2,n)}} \\ \frac{r_{(2,1)}}{r_{(3,1)}} & \frac{r_{(2,2)}}{r_{(3,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(3,n)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{r_{(m-1,1)}}{r_{(m,1)}} & \frac{r_{(m-1,2)}}{r_{(m,2)}} & \cdots & \frac{r_{(m-1,n)}}{r_{(m,n)}} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4}$$

En la Ecuación 5 se muestra la matriz de correlaciones de los vecinos del este:

$$E = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(1,2)}} & \frac{r_{(1,2)}}{r_{(1,3)}} & \cdots & \frac{r_{(1,n-1)}}{r_{(1,n)}} & 0 \\ \frac{r_{(2,1)}}{r_{(2,2)}} & \frac{r_{(2,2)}}{r_{(2,3)}} & \cdots & \frac{r_{(2,n-1)}}{r_{(2,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{r_{(m,1)}}{r_{(m,2)}} & \frac{r_{(m,2)}}{r_{(m,3)}} & \cdots & \frac{r_{(m,n-1)}}{r_{(m,n)}} & 0 \end{bmatrix} \tag{5}$$

En la Ecuación 6 se tiene la matriz de correlaciones de los vecinos de oeste:

$$W = \begin{bmatrix} 0 & \frac{r_{(1,2)}}{r_{(1,1)}} & \frac{r_{(1,3)}}{r_{(1,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(1,n-1)}} \\ 0 & \frac{r_{(2,2)}}{r_{(2,1)}} & \frac{r_{(2,3)}}{r_{(2,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(2,n-1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{r_{(m,2)}}{r_{(m,1)}} & \frac{r_{(m,3)}}{r_{(m,2)}} & \cdots & \frac{r_{(m,n)}}{r_{(m,n-1)}} \end{bmatrix} \tag{6}$$

La Ecuación 7 muestra la matriz de correlaciones A para los vecinos del noroeste:

$$A = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & \frac{r_{(2,2)}}{r_{(1,1)}} & \frac{r_{(2,3)}}{r_{(1,2)}} & ... & \frac{r_{(2,n)}}{r_{(1,n-1)}} \\ 0 & \frac{r_{(3,2)}}{r_{(2,1)}} & \frac{r_{(3,3)}}{r_{(2,2)}} & ... & \frac{r_{(3,n)}}{r_{(2,n-1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{r_{(m,2)}}{r_{(m-1,1)}} & \frac{r_{(m,3)}}{r_{(m-1,2)}} & ... & \frac{r_{(m,n)}}{r_{(m-1,n-1)}} \end{bmatrix} \tag{7}$$

La Ecuación 8 muestra la matriz de correlaciones B para los vecinos del noreste:

$$B = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \frac{r_{(2,1)}}{r_{(1,2)}} & \frac{r_{(2,2)}}{r_{(1,3)}} & \cdots & \frac{r_{(2,n-1)}}{r_{(1,n)}} & 0 \\ \frac{r_{(3,1)}}{r_{(2,2)}} & \frac{r_{(3,2)}}{r_{(2,3)}} & \cdots & \frac{r_{(3,n-1)}}{r_{(2,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{r_{(m,1)}}{r_{(m-1,2)}} & \frac{r_{(m,2)}}{r_{(m-1,3)}} & \cdots & \frac{r_{(m,n-1)}}{r_{(m-1,n)}} & 0 \end{bmatrix} \tag{8}$$

En la Ecuación 9 se muestra la matriz de correlaciones C para los vecinos del suroeste:

$$C = \begin{bmatrix} 0 & \frac{r_{(1,2)}}{r_{(2,1)}} & \frac{r_{(1,3)}}{r_{(2,2)}} & \cdots & \frac{r_{(1,n)}}{r_{(2,n-1)}} \\ 0 & \frac{r_{(2,2)}}{r_{(3,1)}} & \frac{r_{(2,3)}}{r_{(3,2)}} & \cdots & \frac{r_{(2,n)}}{r_{(3,n-1)}} \\ 0 & \vdots & \vdots & \ddots & \vdots \\ \vdots & \frac{r_{(m-1,2)}}{r_{(m,1)}} & \frac{r_{(m-1,3)}}{r_{(m,2)}} & \cdots & \frac{r_{(m-1,n)}}{r_{(m,n-1)}} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{9}$$

Y finalmente, en la Ecuación 10 se muestra la matriz de correlaciones D para los vecinos del sureste:

$$D = \begin{bmatrix} \frac{r_{(1,1)}}{r_{(2,2)}} & \frac{r_{(1,2)}}{r_{(2,3)}} & ... & \frac{r_{(1,n-1)}}{r_{(2,n)}} & 0 \\ \frac{r_{(2,1)}}{r_{(3,2)}} & \frac{r_{(2,2)}}{r_{(3,3)}} & ... & \frac{r_{(2,n-1)}}{r_{(3,n)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \frac{r_{(m-1,1)}}{r_{(m,2)}} & \frac{r_{(m-1,2)}}{r_{(m,3)}} & ... & \frac{r_{(m-1,n-1)}}{r_{(m,n)}} & \vdots \\ 0 & 0 & 0 & ... & 0 \end{bmatrix} \tag{10}$$

Adicionalmente, si $\sigma_{M(i,,j)}$ es la correlación del nodo ubicado en la fila i y en la columna j correspondiente a la matriz M. Se puede estimar el valor de un nodo $\hat{r}_{(i,j)}$ como un promedio de lecturas y/o estimaciones anteriores de sus vecinos. Por lo tanto, para cada nodo (i, j), el valor estimado está dado por la Ecuación 11:

$$\hat{r}_{(i,j)} = \frac{\begin{array}{c}\sigma_{A(i,j)}r_{(i-1,j-1)} + \sigma_{N(i,j)}r_{(i-1,j)} + \sigma_{B(i,j)}r_{(i-1,j+1)} + \sigma_{W(i,j)}r_{(i,j-1)} + \\ \sigma_{E(i,j)}r_{(i,j+1)} + \sigma_{C(i,j)}r_{(i+1,j-1)} + \sigma_{S(i,j)}r_{(i+1,j)} + \sigma_{D(i,j)}r_{(i+1,j+1)}\end{array}}{L} \quad (11)$$

Donde L es el número de vecinos para el cual una lectura y/o estimación ya se conoce, y adicionalmente $r_{(i,j)} = 0$ para valores de i,j que no pertenezcan a los intervalos $1 \leq i \leq m$ y $1 \leq j \leq n$.

Evidentemente, para estimar el valor de un nodo se requiere saber el valor de uno de sus vecinos adyacentes tratándose de la topología que ya se describió anteriormente. Sin embargo, si se conocen de antemano una mayor cantidad de lecturas de los vecinos se anticipa que el valor resultante de la estimación es más preciso. De lo contrario, si no se cuenta con la lectura ni la estimación de alguno de los nodos vecinos, la estimación de la lectura no es posible bajo este esquema.

Un factor importante a considerar para el desarrollo del algoritmo de infección para WSN es el valor del umbral de infección λ, el cual se define como el mínimo número de vecinos requeridos (es decir, el valor del nodo se debe conocer ya sea por una estimación anterior o por una lectura física) para llevar a cabo la estimación de la lectura de un nodo dado. Este parámetro se puede asignar arbitrariamente, sin embargo, su valor dicta qué tan rápido se da el proceso de infección en la red de sensores. Esto se debe a que se puede establecer intuitivamente que con un valor pequeño de λ, la infección se propaga más rápidamente y por lo tanto el número de operaciones requeridas para el algoritmo es menor, pero si el valor del umbral es pequeño también se sacrifica precisión en las estimaciones. En contraste, si se asigna un valor grande de λ, se ocupa un mayor número de iteraciones para que el proceso de infección se propague por toda la red, pero consecuentemente, se obtienen estimaciones más precisas en los nodos. Los

posibles valores de λ deben ser entre 1 y el número máximo de vecinos con los que cuenta un nodo (ocho en el caso de la topología ya presentada).

En la Figura 11 se presenta un ejemplo que muestra cómo se propaga la infección en una red de sensores conformada por una malla de 10×10 nodos de sensado, en este ejemplo se fija el valor de λ a 2, indicando que se requiere de conocer los valores de al menos dos de los nodos vecinos para poder llevar a cabo la estimación del nodo actual. Nótese que en este caso en particular solo se requieren tres iteraciones del algoritmo de infección para poder infectar todos los nodos de la red.

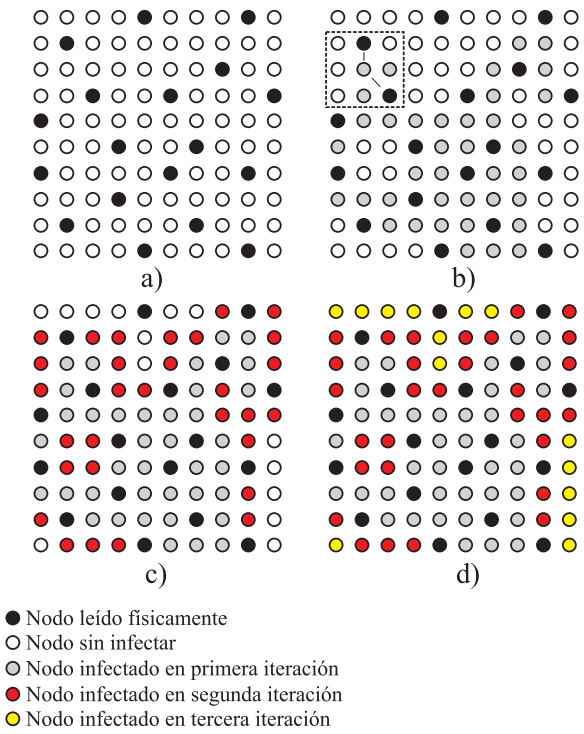

Figura 11. Ejemplo del proceso de infección en una WSN.

En el ejemplo mostrado en la Figura 11, como se puede observar en la Figura 11d, toda la red de sensores se infectó, lo cual implica que todos los valores de los nodos se conocen o se estimaron. Sin embargo, esto no siempre es el caso, ya que dependiendo del valor de λ y de la posición de los nodos leídos inicialmente, se podría llegar a tener

una red que aún cuente con nodos *Inmunes* al final de la ejecución del algoritmo. Por tal motivo, se tiene que encontrar un valor de λ y un criterio de selección apropiados que minimicen el número de los nodos inmunes. El valor del criterio de selección para este parámetro y el criterio de selección de los nodos leídos inicialmente no se estudiaron a fondo en esta parte del trabajo y representan un tema interesante de investigación en el área.

IV.4 Experimentos y simulaciones

En esta sección se presentan los resultados obtenidos a través de una serie de experimentos y simulaciones que se llevaron a cabo para evaluar el algoritmo de diseminación propuesto. A través de estos experimentos, se mide la precisión de las estimaciones realizadas al comparar con los valores reales de los datos recolectados en la aplicación de monitoreo; también se estimaron los ahorros de energía generados a partir de la reducción de las operaciones de comunicaciones.

IV.4.1 Experimentos

Los experimentos se llevaron a cabo en un invernadero donde se cultivan plantas de tomate. El invernadero tiene unas dimensiones de 22 metros de ancho, 8 metros de largo y 4 metros de alto. Se colocaron nueve nodos de sensado dentro del invernadero (nodos MicaZ con un módulo de sensores modelo MTS310). Idealmente, hubiera sido mejor el realizar los experimentos con una red más densamente poblada pero desafortunadamente no fue posible debido a la falta de recursos. Sin embargo, en la sección de simulaciones se presentan los resultados obtenidos con redes más densas.

Se capturaron lecturas de temperatura en el invernadero durante un período de

dos horas aproximadamente con una frecuencia de muestreo de une lectura cada ocho segundos. Se llevaron a cabo varios experimentos cambiando el número de sensores que a los cuales se les solicitó en forma explícita su lectura; el criterio de selección que se aplicó era el de seleccionar primero a los nodos que reportaban un mayor poder de energía remanente en sus baterías. El algoritmo se ejecutó utilizando un valor de 2, luego un valor de 3, así hasta 9 nodos (con 9 nodos no hay error ya que se solicitan las lecturas a todos los nodos).

En la Figura 12 se presenta una gráfica que contiene el error cuadrático medio y los ahorros de energía (en términos de porcentaje) resultantes de los experimentos realizados al algoritmo propuesto. El error se calculó al comparar los datos leídos del campo de sensado con las estimaciones hechas por el algoritmo. El porcentaje de energía ahorrada se representa por cuántos nodos se utilizaron para procesar y comunicar datos, por ejemplo, un 90% en ahorro significa que solo el 10% de los nodos se utilizaron para leer, calcular y transmitir sus valores y el resto se mantuvieron dormidos. Nótese que como se hubiera esperado, con un número mayor de nodos seleccionados se minimizaría el error cuadrático medio de las estimaciones pero el ahorro en energía también sería mínimo en comparación a si se selecciona un número menor de nodos.

Otro factor importante observado en los experimentos es que para un grupo mayor de nodos seleccionados (mayor a cuatro en el caso presentado) solo se requiere un pequeño número de iteraciones (solo uno en este caso) en comparación a si solo se selecciona un pequeño grupo de nodos (menos de cuatro en el caso presentado), en el cual se requeriría de dos iteraciones del algoritmo de infección.

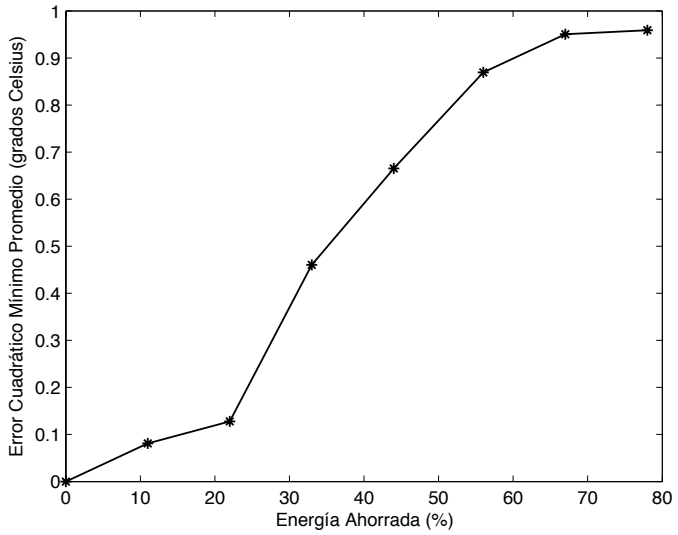

Figura 12. Gráfica del porcentaje de energía ahorrada por el algoritmo contra el error cuadrático medio de las estimaciones resultantes.

IV.4.2 Simulaciones

Las simulaciones se llevaron a cabo considerando 100 nodos de sensado en una malla de 10×10. Los valores utilizados en las simulaciones se generaron por medio de una función pseudoaleatoria; con valores de entre 26.018 y 36.569 grados Celsius (en concordancia con los valores observados en los experimentos del invernadero).

El algoritmo propuesto se programó en Matlab y sus valores iniciales fueron arbitrariamente asignados del conjunto de valores generados por la función pseudoaleatoria. Las principales tareas desarrolladas en la simulación fueron:

- Primero, el programa calcula las matrices de correlación de los valores de temperatura generados.

- Enseguida, se selecciona un subconjunto de nodos. En este caso se hizo una selección de nodos arbitraria; sin embargo, en una implementación real se debe utilizar algún criterio bien definido (como el nivel de energía remanente por ejem-

plo).

- Finalmente, se estiman los valores de los nodos restantes en el campo de sensado.

De manera similar a como se hizo en los experimentos, para determinar la precisión de los valores estimados, éstos se compararon con el conjunto de datos de temperatura generados aleatoriamente. Se ejecutó el algoritmo con valores diferentes de λ y seleccionando diferentes números de nodos para simular las lecturas físicas.

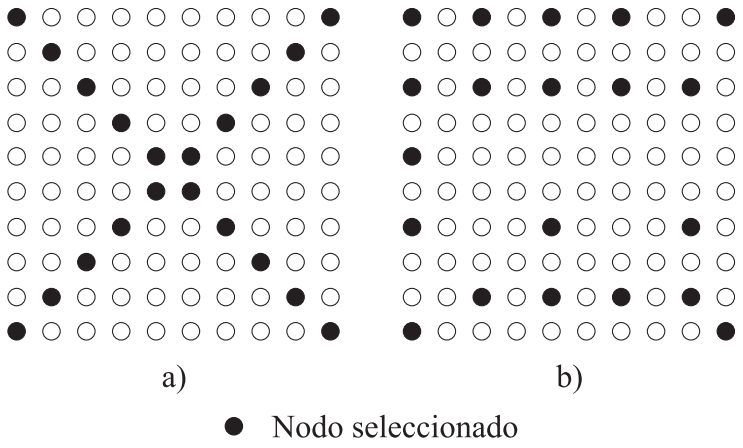

Figura 13. Patrones de nodos seleccionados en las simulaciones.

En la primera simulación se seleccionó un subconjunto de 20 nodos que siguen un patrón que forman dos líneas diagonales a lo largo de la malla (ver Figura 13a), y se ejecutó el algoritmo para valores de λ de 2, 3 y 4.

La Figura 14 muestra que con un valor pequeño de λ se requiere de pocas iteraciones del algoritmo para realizar el total de las estimaciones. Sin embargo, intuitivamente se puede establecer que con un valor grande de λ se pueden obtener estimaciones más precisas debido al hecho de que se utiliza una mayor cantidad de información de los vecinos para realizar cada estimación. Una de las desventajas de utilizar un valor grande de λ es el hecho de que existe un límite superior para el cual ya no se pueden estimar todos los valores del campo de sensado. En el caso de esta simulación, el límite superior

es de 4, debido a que con el patrón que se utilizó para la selección de nodos, no hay nodo alguno en la malla que cuente con 4 vecinos con sus valores conocidos. Por tal motivo, en este caso no se puede estimar valor alguno.

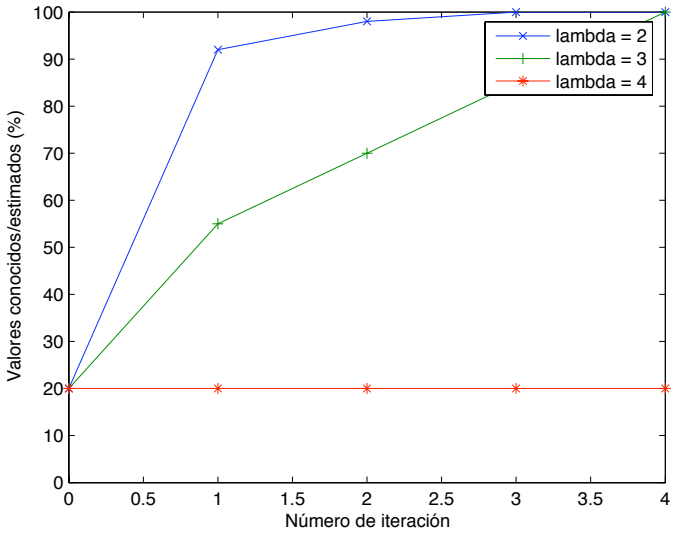

Figura 14. Gráfica del porcentaje de valores estimados para la primera simulación.

En la segunda simulación, se seleccionó el mismo número de nodos (20), pero se utilizó un conjunto de nodos mejor ubicados estratégicamente (ver Figura 13b), se ejecutó el algoritmo utilizando los mismos valores de λ que en la simulación anterior. En la Figura 15 se muestra que, en contraste con la simulación anterior, con un valor de λ de 4 se puede estimar todo el campo de sensado, la desventaja de este escenario es que el algoritmo estima los valores más lentamente (requiere de un mayor número de iteraciones), pero a cambio, se puede establecer que se obtienen estimaciones más precisas. Nótese que las curvas de $\lambda = 2$ y $\lambda = 3$ están sobrepuestas (esto significa que el algoritmo estima los valores con el mismo número de iteraciones), de tal forma que este caso en particular no existe un costo computacional adicional al aumentar el valor de λ de 2 a 3, por tal motivo, sería más efectivo el utilizar el valor más alto (3 en este

caso).

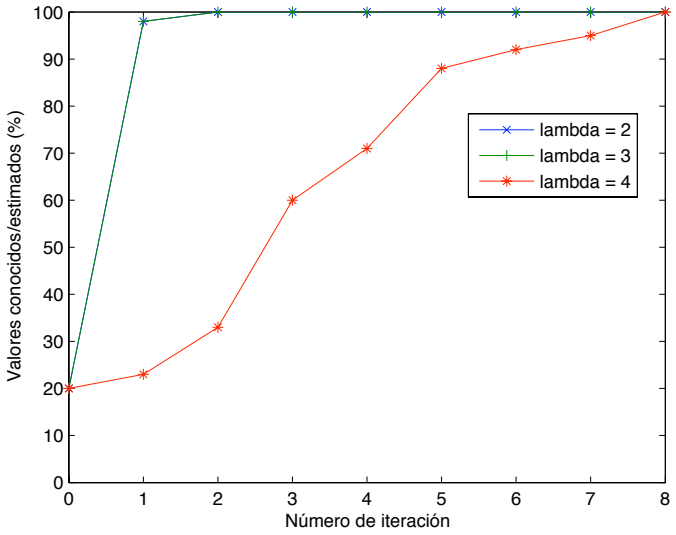

Figura 15. Gráfica del porcentaje de valores estimados para la segunda simulación.

Finalmente, después de observar los resultados de los experimentos y de las simulaciones, se puede concluir que el seleccionar los parámetros de nuestro algoritmo (seleccionar el subconjunto de nodos y el valor de λ) es un problema de optimización complejo que involucra precisión, costo computacional y energía requerida para realizar búsquedas en la red de sensores. Este problema no se abordó en el trabajo presentado en este capítulo, sin embargo, representa una oportunidad interesante de investigación en un futuro.

En este capítulo se presentó una propuesta de adaptación del algoritmo de infección para la diseminación de datos en redes inalámbricas de sensores, como se mostró, dicha propuesta puede generar ahorros significativos en el consumo de energía; algunos detalles tales como el criterios de selección de nodos y la determinación del valor del parámetro λ se dejaron como problemas abiertos, los cuales podrían dar una idea

más exacta acerca de los beneficios que se pueden esperar con la aplicación de dicho enfoque. En el siguiente capítulo se presentan dos propuestas originales para la diseminación de datos desde los nodos de sensado al nodo líder, en ambas propuestas se tiene como premisa la eliminación de datos redundantes por las razones que ya se han mencionado en este capítulo. Adicionalmente, las propuestas que se presentan en el siguiente capítulo integran servicios de seguridad que algunos escenarios de aplicación determinados pudieran demandar.

Capítulo V

Redundancia y Diseminación Eficiente

V.1 Introducción

Recientemente se han publicado muchas aplicaciones para redes inalámbricas de sensores (WSN) formadas por cientos o inclusive miles de nodos (Tolle *et al.*, 2005; Allred *et al.*, 2007; Gandhi *et al.*, 2007; Basha *et al.*, 2008); por tal motivo, la alta escalabilidad es un requerimiento fundamental para este tipo de redes. Por otra parte, las redes de malla no cumplen con este requerimiento ya que cada nodo desempeña el mismo papel y tiene la responsabilidad de sensar el ambiente y la de enrutar tráfico dirigido hacia otros nodos y/o la estación base. En contraste, el agrupamiento jerárquico de nodos (del término en inglés "clustering") es un enfoque que se ha utilizado para lograr eficiencia energética y proporcionar escalabilidad a las WSN (Banerjee y Khuller, 2001). Este agrupamiento facilita la distribución de control sobre la red y por lo tanto permite una comunicación más localizada. Los nodos que pertenecen a un cluster (grupo de nodos) específico pueden interactuar únicamente de manera directa con un nodo predeterminado que desempeña el papel del líder del cluster (del término en inglés "clusterhead"). Solo los nodos líderes necesitan comunicarse a distancias más grandes para poder llegar hacia la estación base; sin embargo, este problema puede resolverse utilizando agrupamiento jerárquico (aplicando agrupamiento jerárquico de manera recursiva). Adicionalmente, las tareas relacionadas a estas redes jerárquicas se realizan

en forma más distribuida en comparación a otro tipo de redes que cuentan con una estructura plana. Por ejemplo, el sensar el ambiente y el comunicar las lecturas al nodo líder son tareas que corresponden a los nodos de sensado mientras que el nodo líder tiene la responsabilidad de procesar los datos sensados de los nodos de su grupo (por medio de agregación de datos) y enviar los resultados hacia la estación base o hacia un nodo de mayor jerarquía en la red.

Un problema que se aborda en el presente documento es el de transmitir los datos sensados por los nodos hacia el nodo líder. Este problema podría sonar trivial al inicio, pero como ya se mencionó anteriormente muchas aplicaciones requieren de redes densamente pobladas para monitoreo ambiental. Este requerimiento se debe principalmente al hecho de que este tipo de nodos son altamente susceptibles a fallas y el ambiente donde se instalan es hostil en la mayoría de los casos. Además, los nodos no cuentan con ningún tipo de protección física, esto con el fin de mantener su bajo costo de producción. Debido al hecho de que el colocar varios nodos en la misma área de cobertura es una práctica muy común, este tipo de redes frecuentemente genera altos niveles de datos redundantes y a su vez esto resulta en excesivos consumos de energía al comunicar los datos de los nodos de sensado hacia el nodo líder del cluster.

En esta propuesta se introduce un mecanismo de transmisión de datos desde un nodo hacia su nodo líder, aprovechando los altos niveles de redundancia para reducir el consumo de energía. Como servicios de valor agregado se incorporan funciones de integridad y seguridad al protocolo propuesto. Como se muestra posteriormente, la incorporación de las funciones de seguridad no incurren en sobrecosto de energía, de hecho, en algunos casos se producen ahorros significativos en su consumo, lo cual eventualmente incrementa el tiempo de vida de la red de sensores.

V.2 Propuesta de Protocolo

V.2.1 Diseño

El protocolo propuesto se basa en un proceso original de autentificación en dos fases, este proceso detecta datos redundantes al mismo tiempo que proporciona servicios de autentificación (Palafox y García-Macías, 2008). Además, otros servicios de seguridad tales como la confidencialidad se pueden proporcionar trivialmente por medio de la misma llave secreta compartida por todo el cluster. Concretamente, el protocolo propuesto se basa en la siguiente premisa (Schneier, 1996), si:

$$A = MAC_f\left(K_1, M_1\right) \tag{12}$$

y

$$B = MAC_f\left(K_2, M_2\right) \tag{13}$$

entonces

$$(A = B)\ \forall\ (K_1, M_1) = (K_2, M_2), f \tag{14}$$

La Ecuación 12 dice que si A es igual al código MAC calculado con la función f y utilizando como entradas la llave secreta K_1 y el mensaje M_1 y en forma similar la Ecuación 13 dice que si B es igual al código MAC utilizando la misma función MAC f que en la Ecuación 12 y utilizando como entradas la llave K_2 y el mensaje M_2; entonces, se dice en la Ecuación 14 que A va a ser igual a B para todo par (K_1, M_1) que sea igual a otro par (K_2, M_2) siempre y cuando se utilice la misma función f. Por lo tanto, si la misma llave la comparten todos los miembros del cluster, el nodo líder es capaz de detectar datos redundantes con solo recibir los códigos MAC de todos ellos.

Para el cálculo del código MAC se selecciona el algoritmo CBC-MAC (*Cipher Block*

Chaining Message Authentication Code) (Bellare *et al.*, 2000; ISO/IEC 9797, 1989). Un aspecto importante a considerar del CBC-MAC así como de otras funciones MAC es el hecho de que es posible que se presenten colisiones, es decir, dos mensajes diferentes pueden generar el mismo código MAC al utilizar la misma llave; la presencia de colisiones afecta negativamente el funcionamiento del protocolo propuesto al detectar datos redundantes cuando esto no necesariamente sea cierto (falsos positivos). Sin embargo, la probabilidad de colisiones del algoritmo CBC-MAC para dos mensajes con longitud m que utilizan un código MAC de longitud n y la misma llave, está acotada por arriba como se muestra en la Desigualdad 15 (Black y Rogaway, 2005):

$$P_n(m) \leq \frac{(2m/n)^2}{2^n} \tag{15}$$

Por lo tanto, la probabilidad de colisión de una implementación prototipo que se aborda en el siguiente capítulo (con mensajes de 96 bits y códigos MAC de 32 bits) es muy pequeña. Esta probabilidad se muestra en la desigualdad 16.

$$P_{32}(96) \leq 8.3819^{-9} \tag{16}$$

Se consideró la posibilidad de integrar un mecanismo basado en correlación espacial para reducir aún más la probabilidad de colisión, pero se decidió que el incremento en los costos de memoria y procesamiento son injustificados ante un escenario tan poco probable[1].

Intercambio de mensajes

En cuanto al intercambio de mensajes se refiere, en el protocolo se consideran cinco tipos de mensajes. En la Tabla I se enlistan esos mensajes así como una breve descripción de

[1]En el peor de los casos se espera un falso positivo cada 12.5 millones de períodos.

su carga útil y su longitud. Se asignaron identificadores numéricos (IDs) en cada tipo de mensaje para su posterior uso en la implementación.

Tabla I. Tipos de mensajes definidos en el protocolo.

Valor ID	**Carga útil (bytes)**	**Descripción**
0x4a	Código MAC (4)	Código MAC enviado hacia el nodo líder
0x4b	Lista de nodos (1)	Petición de datos enviada del nodo líder hacia los nodos de sensado
0x4c	Lecturas de sensado (20)	Lecturas de sensores enviadas hacia el nodo líder
0x4d	Vacío (0)	Solicitud de retransmisión a un nodo específico
0x4e	CNTR ||MAC (6)	Paquete de actualización del contador

Para la implementación del prototipo se consideraron paquetes con un encabezado de 10 bytes, tal y como lo establece el estándar IEEE 802.15.4 (IEEE, 2003) con un tamaño máximo de paquete de 30 bytes (20 bytes de carga útil) como en TinyOS (Hill *et al.*, 2000).

Los cinco tipos de mensaje definidos se utilizan en diferentes fases del protocolo. Enseguida se describe el funcionamiento del protocolo propuesto:

1. Los nodos de sensado que pertenecen al cluster sensan el ambiente, los nodos líder se pre-determinaron por medio de algún algoritmo de selección de líder tal como

LEACH (Heinzelman *et al.*, 2002) (Low Energy Adaptive Clustering Hierarchy).

2. Cada nodo calcula su código MAC de 4 bytes utilizando una llave secreta compartida con el cluster $K_{cluster}$, sus lecturas obtenidas por los sensores ($DATA$) y un contador ($CNTR$) que está sincronizado con el resto del cluster.

$$Nodo \longrightarrow NodoLider : MAC(K_{cluster}, DATA \parallel CNTR)$$

3. Cuando todos los códigos MAC se han recibido (o tras haber transcurrido un tiempo límite, *timeout*), el nodo líder los almacena en un buffer de memoria y los clasifica de acuerdo al nodo del cual provienen (nodo fuente).

4. Posteriormente, el nodo líder detecta cuáles códigos MAC se encuentran repetidos dentro del buffer, y consecuentemente, infiere cuáles nodos tienen información redundante.

5. Una vez que los nodos con información redundante se han identificado, el nodo líder construye una lista de nodos ($LIST$) para los cuales una petición específica de datos es necesaria.

6. Cuando la lista se haya construido, se anexa a uno o más paquetes de tipo *0x4b* (dependiendo del tamaño de la lista). Estos paquetes se envían mediante transmisión tipo *broadcast* del nodo líder a los nodos de sensado. Los paquetes se autentifican con el contador que se mencionó anteriormente (paso 3) con el fin de brindar actualidad de los datos (*data freshness*) y así prevenirse contra ataques de "almacenamiento y reenvío". Como se muestra posteriormente, este método es el más simple ya que no involucra de otras capas, pero no es el más eficiente. Una forma más eficiente de lograr esto es el utilizar algunos bits en las señales

ACK de la capa de Acceso al Medio para indicar a cada nodo que sus lecturas son requeridas por el nodo líder.

$$NodoLider \Rightarrow Nodos : LIST, MAC(K_{cluster}, LIST \parallel CNTR)$$

7. Los nodos líder a los que se les solicita información, deben integrar un paquete *0x4c* con las lecturas de sus sensores, y este paquete se debe enviar hacia el nodo líder:

$$Nodo \rightarrow NodoLider : DATA$$

8. Finalmente, el nodo líder ya conoce las lecturas de los nodos de los cuales recibió respuesta a sus peticiones, así como también conoce aquellas lecturas pertenecientes a los nodos con códigos MAC duplicados; por lo tanto, en este punto la verificación de la integridad y/o autenticidad de los mensajes la lleva a cabo el nodo líder. Si esta verificación falla, el nodo líder puede nuevamente solicitar los datos a tales nodos. De forma alternativa, un algoritmo de estimación (Palafox y García-Macías, 2006) se puede utilizar en lugar de solicitar retransmisión.

Sincronización del contador

La seguridad del protocolo propuesto recae fuertemente en qué tan oculto se mantenga el contador compartido por todos los miembros del cluster. Además, para que el protocolo funcione adecuadamente, el contador sincronizarse dentro del cluster, es decir, el valor interno del contador debe ser el mismo en todos los nodos miembro del cluster. Debido a posibles desvíos en el reloj interno de los nodos (debido a imprecisiones en el oscilador de cristal), es posible que la sincronización se rompa durante la operación de la red. Por tal motivo, se está proponiendo una técnica simple de sincronización del contador para evitar que se rompa ésta. Esta técnica la coordina el nodo líder,

y consiste en que éste envíe paquetes de actualización (de tipo *0x4e*) periódicamente. Estos paquetes contienen el nuevo valor del contador, el cual lo genera aleatoriamente con una distribución uniforme el nodo líder, cuando el nodo miembro recibe un paquete de actualización de contador autentificado, tiene que actualizar el valor de su contador interno al que acaba de recibir.

El restablecer el contador periódicamente a un nuevo valor generado aleatoriamente también dificulta a un atacante el poder estimar su valor produciendo un efecto similar al de las técnicas tradicionales de renovación de llaves. En este caso, se está abordando la sincronización junto con la actualización del contador utilizando[2] la misma técnica.

Por motivos de seguridad, se tienen que encriptar los mensajes de actualización de contador, esto se debe a que si un intruso conoce el valor del contador actual, puede falsificar paquetes y engañar al nodo líder para que éste acepte mensajes de datos y/o códigos MAC.

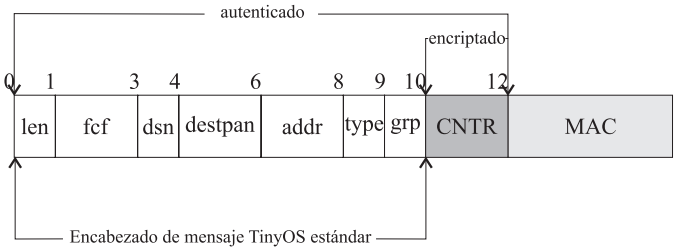

Figura 16. Formato del mensaje de actualización del contador.

Para optimizar memoria de código de programa, para la encriptación, se emplea el mismo encriptador de bloques que se utiliza para el algoritmo de generación de códigos MAC (basado en el encriptador RC5). En la Figura 16 se muestra el formato del paquete de actualización del contador. Como se muestra en el paso 6 del algoritmo presentado,

[2]En este contexto, el contador es secreto, y su valor afecta directamente el valor del código MAC calculado. Por tal motivo se puede ver a la actualización del contador como una alternativa a la renovación de llaves.

el nodo líder envía mediante broadcast un mensaje que contiene el contador encriptado ($CNTR$) y el código MAC calculado tomando en base el encabezado del mensaje y el $CNTR$ encriptado. Nótese que en este caso, el código MAC se envía junto con el resto del paquete tal y como se hace en el enfoque tradicional.

Esta técnica de sincronización del contador también agrega ciertos requerimientos de recursos adicionales para la red de sensores, pero se prevé que no se tendrán que enviar actualizaciones del contador con mucha frecuencia. La frecuencia de los mensajes de actualización del contador depende directamente de la longitud del periodo de sensado[3] así como de la plataforma de hardware en donde se realice la implementación. En el Capítulo VI (Prototipos y Experimentos) se discuten más detalles acerca de las medidas tomadas en relación a la frecuencia de los mensajes de actualización en el prototipo experimental.

V.2.2 Implementación

Para la implementación del prototipo, se consideró un sistema de monitoreo en una bodega. El objetivo principal de la aplicación es el de proveer un ambiente controlado para el almacenamiento de sustancias químicas y otros materiales altamente sensibles a cambios ambientales. Bajo este escenario de implementación, la seguridad es esencial, debido a que no se pueden permitir ataques tales como cuando un intruso coloca nodos que inyecten lecturas falsas de los sensores en la red de monitoreo.

Como se muestra en la Figura 17, los nodos se colocaron muy cerca uno de otro (la distancia promedio entre nodos era de 2.13 m aproximadamente) dentro de una bodega en forma de "L" relativamente pequeña (43 m^2), y como el nodo líder (CH) se colocó

[3]Se define el periodo como el tiempo transcurrido entre la recepción de dos códigos MAC de un mismo nodo.

en el centro de la bodega, la distancia máxima entre un nodo de sensado y el nodo líder era de 6 m aproximadamente. Por lo tanto, se puede establecer comunicación de un solo salto entre el nodo líder y cada uno de los nodos del cluster.

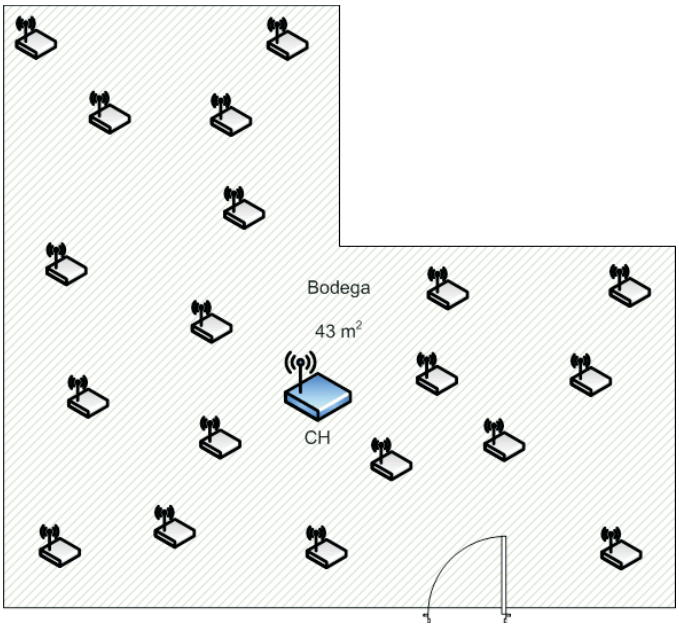

Figura 17. Topología de los nodos de una aplicación de monitoreo en una bodega.

Un aspecto crucial en la implementación del protocolo es decidir qué algoritmos básicos de criptografía se van a utilizar. El algoritmo AES (*Advanced Encryption Standard*)(Daemen y Rijmen, 2000) se utiliza comúnmente en seguridad de redes tradicionales; se consideró la posibilidad de utilizarlo en este protocolo también. Sin embargo, este algoritmo consume 800 bytes en tablas de consulta, un requerimiento que lo hace poco viable considerando la plataforma de implementación. También se analizaron otras alternativas, tales como el algoritmo DES (*Data Encryption Standard*) (National Institute of Standards and Technology, 1999), pero éste requiere mucha memoria para almacenar tablas de permutación. Finalmente, se decidió utilizar el algorimo RC5 (Rivest, 1994), debido a que presenta muchas características que lo hacen apropiado para implementarse en plataformas de recursos con fuertes limitaciones tales como las

redes inalámbricas de sensores. Por ejemplo, requiere de poca memoria, es un algoritmo rápido de procesar y es muy flexible en cuanto a los tamaños del bloque de datos y de las llaves que maneja. En cuanto a los tamaños del bloque de datos y de la llave, se seleccionó RC5-32/12/16 debido a que de acuerdo a la literatura (Rivest, 1994), ofrece un buen equilibrio entre eficiencia y seguridad. Bajo este esquema, el encriptador procesa bloques de 32 bits a la vez durante 12 etapas utilizando una llave criptográfica de 16 bytes. Existen muchas formas de calcular el código MAC para un mensaje dado, entre ellas las funciones HMAC y SHA-1(National Institute of Standards and Technology, 1995). Sin embargo, para implementar alguna de estas funciones se requiere mayor cantidad de memoria para datos y para código de programa. Por tal motivo, como ya se mencionó anteriormente, se seleccionó el algoritmo CBC-MAC. Este algoritmo utiliza un encriptador convencional como RC5 en forma iterativa para calcular el código MAC. Por lo tanto, al seleccionar este algoritmo se utiliza el mismo encriptador de bloques para ofrecer el servicio de confidencialidad y autentificación, lo cual obviamente resulta en un ahorro de memoria (de datos y de código de programa).

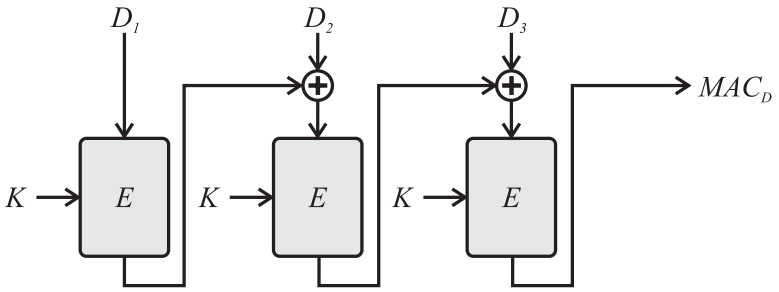

Figura 18. Operación del algoritmo CBC-MAC.

En la Figura 18 se muestra la operación del algoritmo CBC-MAC, en donde cada bloque de datos lo procesa el encriptador y su salida se suma con el siguiente bloque de datos. La salida del último bloque de encriptación es el código MAC del mensaje de entrada D.

El protocolo se implementó con los motes *TMote Sky* y el sistema operativo *TinyOS 1.1.15*, esta implementación se basó en una aplicación que periódicamente envía las lecturas de todos los sensores de los nodos.

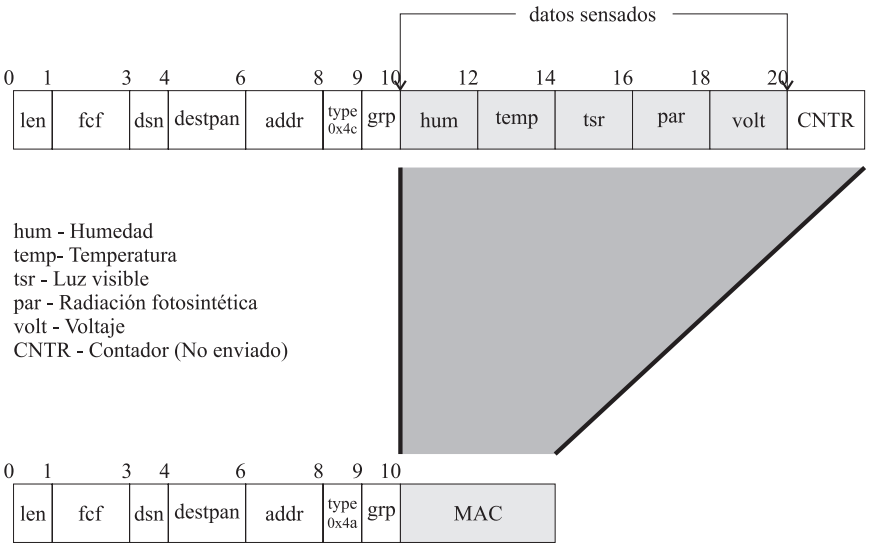

Figura 19. Formato de los mensajes que transportan los datos de los sensores y los códigos MAC.

En la Figura 19 se muestra el formato de los paquetes *0x4a* y *0x4c* (ver Tabla I), los primeros 10 bytes representan el encabezado de un paquete estándar de TinyOS. Además, en la implementación, los datos sensados incluyen lecturas de los sensores de humedad, temperatura, luz visible, radiación fotosintética y el voltaje de la batería a bordo de los motes *TMote Sky*. Como se muestra en esa figura, la longitud de los datos sensados es de 10 bytes y el código MAC tiene una longitud de 4 bytes, los datos de sensado y el contador sincronizado ($CNTR$) se utiliza para calcular el código MAC. Sin embargo, el contador no se envía con el mensaje.

Para solucionar el problema de pequeñas discrepancias entre las lecturas de diferentes nodos debido a la calibración, se utilizan intervalos discretos para agrupar los valores de las lecturas de los sensores, cada intervalo lo representa su valor central, de manera que cuando un valor cae dentro de un intervalo determinado, el nodo transmite

el código MAC calculado a partir del valor central de dicho intervalo. Una desventaja de esta técnica es que se sacrifica cierta precisión pero a cambio se aumenta la probabilidad de obtener información redundante lo cual beneficia el consumo de energía como se muestra en el siguiente capítulo. Por ejemplo, los motes *TMote Sky* tienen un sensor de humedad/temperatura SHT11 de la compañía *Sensirion*[4], este sensor tiene una resolución de 14 bits con una precisión de $\pm 0.4^\circ$ C @ 25° C. De acuerdo a las especificaciones técnicas, la temperatura (en grados Celsius) puede calcularse de la salida digital de 14 bits (SO_T) con la ecuación[5] 17.

$$T = -39.60 + (0.01)(SO_T) \tag{17}$$

De la Ecuación 17 se puede observar que el bit menos significativo de la salida de 14 bits representa solo una centésima parte de un grado Celsius. Por lo tanto, si se relajan los requerimientos de precisión en la temperatura[6] se puede proponer un esquema de intervalos para enfrentarse a posibles problemas de calibración y de pequeñas variaciones locales en nodos dentro del mismo cluster. Específicamente en la aplicación prototipo se consideraron intervalos de 0.32° C, esto se hizo con la siguiente manipulación de bits: se enmascararon (pusieron en cero) los primeros cuatro bits menos significativos y el quinto se puso en uno. Con esto se evitó cualquier operación de punto flotante en el nodo, las cuales no se recomiendan para los nodos debido a su alto costo computacional.

En la Figura 20 se muestran los intervalos generados al utilizar el esquema descrito, el utilizar intervalos de 0.32° C genera un error de hasta $\pm 0.16^\circ$ C, lo cual es admisible en ciertas aplicaciones, entre ellas la aplicación de la bodega propuesta en este capítulo.

[4]Sitio web: http://www.sensirion.com/

[5]Cuando se utiliza una fuente de 3V.

[6]Dado el hecho de que ya se está utilizando un sensor con una precisión de $\pm 0.4^\circ$ C, se puede dar por hecho de que los requerimientos de precisión en la temperatura no son muy estrictos.

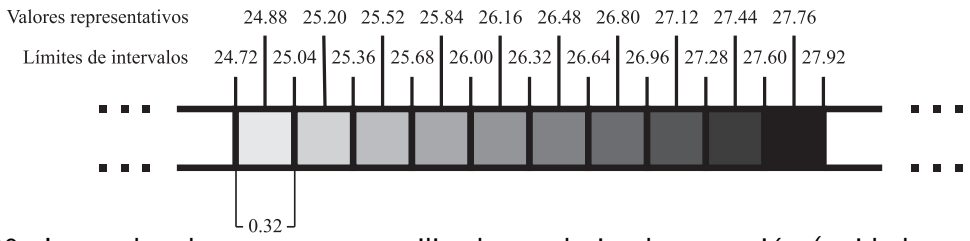

Figura 20. Intervalos de temperatura utilizados en la implementación (unidades en grados Celsius).

Las lecturas provenientes de otros sensores se manejaron en forma similar, solo se varió el ancho del intervalo de acuerdo a los requerimientos de precisión de cada una de las variables sensadas.

V.3 Propuesta para la diseminación de comandos de voz en una red inalámbrica de sensores

Para dar mayor validez a la propuesta de protocolo presentada en la sección anterior, también se diseñó una variante de dicho protocolo que consiste en una arquitectura para la captura de voz por medio de redes de sensores (Palafox y García-Macías, 2009). En esta sección se detalla el procedimiento de implementación que se siguió para construir un prototipo basado en la arquitectura mencionada. Dicho prototipo está orientado a ambientes seguros de cómputo ubicuo en el hogar pero puede extenderse hacia otros escenarios similares. La propuesta de esta arquitectura se basa en reducir los datos redundantes mientras se mantiene la redundancia de nodos para ofrecer confiabilidad, al mismo tiempo se ofrecen servicios básicos de seguridad sin sacrificar una cantidad considerable de recursos. Además se muestra que este enfoque es eficiente en el consumo de energía en comparación al enfoque tradicional de "captura y envía". Otra contribución de este trabajo es que a través de las experiencias en la implementación del sistema, se

provee de una ruta para el desarrollo de aplicaciones eficientes en hogares inteligentes basadas en voz.

V.3.1 Antecedentes

El hogar idealmente provee de un ambiente seguro y confortable en el cual uno puede relajarse, comunicarse, aprender y divertirse (Intille, 2002). Por lo anterior, varios grupos de investigación han dirigido sus esfuerzos en llevar a los hogares las aplicaciones de cómputo pervasivo que se han ido desarrollando, pero que en su fase inicial de desarrollo solo se probaron en laboratorios con ambientes controlados.

El "Aware Home" (Kidd *et al.*, 1999) creado en la Universidad *Georgia Tech* se basa en el desarrollo de un laboratorio para la investigación de cómputo ubicuo aplicado a las actividades de la vida diaria. Otros trabajos se han enfocado en facilitar la interacción de los usuarios con ambientes de hogares inteligentes (Mozer, 1998) a través de la implementación de sistemas residenciales básicos tales como sistemas de aire acondicionado, de iluminación, de ventilación y de calentamiento de agua. Además, se han construido sistemas prototipo en residencias reales (no solamente en laboratorios).

Siguiendo en esta misma línea de trabajo, para el ser humano el habla es una de las formas más simples de interactuar con el ambiente, en este caso, con un ambiente de hogar. Nuevas tecnologías para ambientes de cómputo ubicuo han surgido después del desarrollo del trabajo de investigación que ya se mencionó y algunas de esas tecnologías (como las WSN por ejemplo) pueden mejorar estas experiencias pasadas utilizando ambiente de hogares.

En el área de investigación en WSN, se han publicado muchas aplicaciones que utilizan sensores de audio, tales como: localización (Simon *et al.*, 2004), vigilancia

(Gu *et al.*, 2005), monitoreo geológico (Werner-Allen *et al.*, 2006) y comunicaciones (Vasilescu *et al.*, 2005). Sin embargo, muy pocas de ellas extraen lecturas de audio puro del ambiente. *EnviroMic* (Luo *et al.*, 2007) es una de esas aplicaciones, ésta captura altos volúmenes de audio y lo almacena en su memoria flash local para posteriormente enviarlos bajo demanda de una forma similar a como lo hace *data-mule* (Shah *et al.*, 2003), el cual utiliza un modelo de "almacena y carga". El trabajo propuesto difiere sustancialmente en el sentido de que en este caso la transmisión de datos la inicia el usuario cuando éste emite un comando de voz al ambiente y no bajo demanda por peticiones hechas por la estación base como sucede con *EnviroMic*. Además de esto, la propuesta integra técnicas básicas de seguridad para que la arquitectura cumpla con escenarios que demanden servicios de seguridad.

Como se puede ver en la literatura, la tecnología de WSN se ha utilizado tradicionalmente para el monitoreo de ambientes en exteriores tales como: actividad volcánica (Lees *et al.*, 2008), agricultura de precisión (Liao y Sarabandi, 2006), monitoreo de habitats (Szewczyk *et al.*, 2004) y muchos otros más. Sin embargo, existe poco trabajo hecho en relación a implementaciones reales en ambientes interiores, donde los usuarios puedan interactuar con el ambiente en forma ubicua mediante la red de sensores. En la implementación realizada, el usuario lanza al ambiente un comando de voz relativamente corto, la red de sensores lo captura por medio del mecanismo de recolección de voz propuesto y posteriormente lo envía a la estación base, donde se llevan a cabo las tareas de reconocimiento de voz.

Por lo tanto, se integró una WSN para experimentar de primera mano a qué grado ayudarían a proveer esa "facilidad de implantación" que prometen tradicionalmente, de tal manera que, la contribución principal de este trabajo es la experiencia recabada a través del diseño e implementación de la arquitectura propuesta.

V.3.2 Diseño e implementación

Generalidades

Como ya se mencionó anteriormente, el sistema de captura de voz se basa en una red inalámbrica de sensores, por tal motivo, la restricción más importante es la limitante de energía; la fuente de energía para los nodos que se utilizaron en la implementación (los motes MicaZ con TinyOS(Hill *et al.*, 2000) mostrados en la Figura 21, cuyas especificaciones se presentan en el Apéndice B) es un par de baterías AA, las cuales se pueden agotar en un período de tiempo relativamente corto. Por lo tanto, la integración de técnicas eficientes en el consumo de energía representan un factor importante que guía el diseño de los protocolos de comunicación para este tipo de plataformas.

Figura 21. Mote MicaZ.

La confiabilidad en la red de captura de comandos de voz es un requerimiento extremadamente importante, y considerando el hecho de que los nodos de sensado son muy susceptibles a fallas, se decidió implantar nodos en forma redundante, es decir, se colocaron varios nodos muy cerca el uno del otro generando áreas de monitoreo que se traslapan. Al hacer esto, si un nodo falla aleatoriamente se cuenta con otros que pueden llevar a cabo las funciones de captura y transmisión sin interrumpir el funcionamiento

global del sistema, asegurando así un proceso de captura de voz confiable. Sin embargo, el colocar nodos en forma redundante trae consigo una gran desventaja: se generan datos redundantes en la red ya que cabe la posibilidad de que dos o más nodos se encuentren capturando el mismo comando de voz simultáneamente, lo cual a la vez, introduce tráfico adicional a la red. Este problema se torna aún más crítico cuando se habla de capturar grandes volumenes de datos tal y como en el caso de aplicaciones de captura de audio digital: el grabar audio aunque sea por tan solo un período corto de tiempo genera grandes cantidades de datos, agotando el ancho de banda y la memoria disponible, ambos recursos ya de por sí limitados. Por citar un ejemplo, si se utiliza una frecuencia de muestreo de 8.192 kHz y solo se graban tres segundos de audio con muestras de 8 bits, se generan un poco más de 24 KB de datos; considerando que el tamaño de un mensaje estándar en TinyOS es de 30 bytes, se tienen que transmitir más de 800 mensajes para transferir esos tres segundos de audio a la estación base. Por lo tanto, el capturar y transmitir el mismo comando de voz por dos o más nodos es simplemente inaceptable.

Los nodos que se encuentran dentro de la misma área pertenecen al mismo grupo (cluster); en cada cluster existe un nodo especial que coordina las actividades del cluster (nodo líder). En la Figura 22, se muestra el diagrama (con algunas fotos) de la implementación prototipo en la cual cada cluster captura los comandos de voz correspondientes a una habitación de la casa. Cada nodo líder recolecta muestras de audio de los nodos de su cluster y las reenvía hacia la estación base representada por una computadora personal.

Para capturar señales de audio del ambiente, los nodos de sensado cuentan con un micrófono de alta sensibilidad. Este micrófono se utiliza inicialmente para detectar la presencia de audio en el ambiente. Para lograr esto, mientras se encuentran en estado

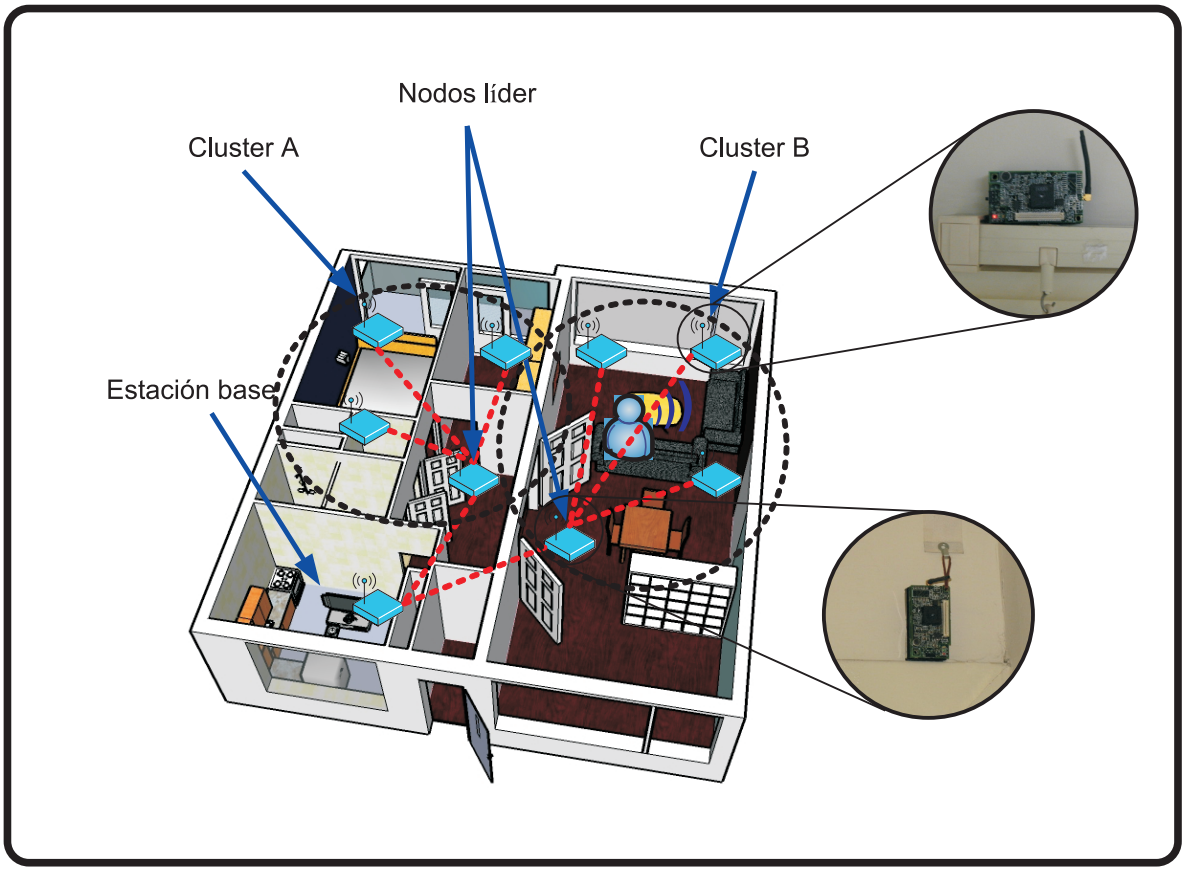

Figura 22. Implementación prototipo de la red de captura de audio.

activo, los nodos continuamente sensan señales de audio con una frecuencia de muestreo de 2 kHz; si la intensidad de la señal sensada rebasa un umbral ya predeterminado, el nodo envía una notificación a través de su interfaz de comunicación inalámbrica hacia el nodo líder; cabe señalar que es posible que el nodo líder reciba más de una notificación de diferentes nodos dentro del mismo cluster, por lo tanto, éste selecciona qué nodo está a cargo de capturar y transmitir el audio. El nodo seleccionado recibe un comando de captura del nodo líder, cuando esto suceda el nodo entra en un modo de *muestreo de alta frecuencia* (*HFS* por sus siglas en inglés) y captura tres segundos de audio[7] y lo almacena en su memoria EEPROM. Cuando el nodo termina de muestrear, sale del modo *HFS* y transfiere el audio capturado al nodo líder, el cual a su vez lo reenvía a la estación base, donde un sistema con mayores recursos de hardware realiza tareas de reconocimiento de voz.

El modo de *muestreo de alta frecuencia* implica el apagar la interfaz de comunicaciones y el muestrear señales de audio por medio del micrófono a una frecuencia de muestreo de 8 kHz. Es absolutamente necesario apagar la interfaz de comunicaciones para poder muestrear a tal frecuencia debido a que las estrictas limitantes de hardware de la plataforma impiden que ambos componentes puedan ser habilitados simultáneamente.

El micrófono en los motes está conectado físicamente a un convertidor analógico digital (ADC) que cuenta con una resolución de 10 bits. Como se mencionó anteriormente, se utilizaron muestras de 8 bits, es decir, no se consideraron los 2 bits menos significativos proveídos por las muestras del ADC. Esto permite simplificar la implementación utilizando una sola variable de 8 bits para cada muestra, adicionalmente,

[7]Los comandos de voz son relativamente cortos, por tanto, se decidió que tres segundos de voz es suficiente.

también se redujeron la cantidad de memoria requerida y el "overhead" de comunicación. Al truncar las muestras de 10 a 8 bits se degrada la calidad del audio, pero como se muestra posteriormente, esto no es significante en la implementación prototipo que se presenta. Al mantener el tamaño de la muestra en 8 bits se pueden incluir 30 muestras en cada mensaje TinyOS como se puede observar en el datagrama mostrado en la Figura 23.

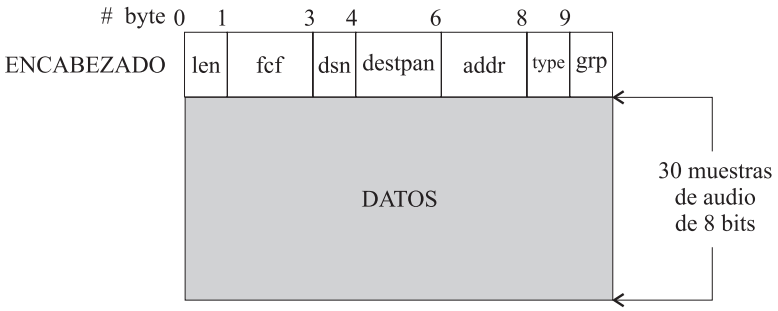

Figura 23. Datagrama del mensaje que transporta las muestras de audio.

Criterio de selección de nodos

Es muy recomendable que el nodo líder alterne la selección de nodos en el cluster, balanceando así la carga de trabajo con el fin de prevenir el agotamiento prematuro de la fuente de energía en ciertos nodos debido a las limitantes de recursos ya mencionadas. Además, para una implementación a nivel de producto terminado, es importante el integrar alguna técnica de rotación de nodo líder, para que de esta manera se extienda el tiempo de vida de la red; ya existe mucho trabajo que se puede encontrar en la literatura relacionado con esto (Heinzelman *et al.*, 2002; Su y Zhang, 2006; Kim *et al.*, 2008). Sin embargo, en la implementación de prueba presentada, el nodo líder ya está predeterminado; esto se debe a que el interés principal es el de evaluar el desempeño de la aplicación de comandos de voz.

Como ya se describió previamente, las muestras de audio se almacenan en la memoria EEPROM. Una justificación adicional para colocar nodos de sensado en forma redundante dentro de una misma área es que el número de operaciones de lectura/escritura en una memoria tipo EEPROM es limitado.

Debido a esto, se está proponiendo un protocolo de recolección en el cual la meta principal es la de eliminar la redundancia de datos. La idea detrás de este esquema es que cada nodo de sensado tiene que enviar un mensaje de reporte de evento a su nodo líder, el contenido de este mensaje es un contador encriptado (por razones de seguridad como se muestran posteriormente). Después, el nodo líder selecciona qué nodo tiene que capturar y enviar el audio a través de la red. Conforme el nodo líder reciba mensajes de audio, éste tiene que reenviarlas hacia una estación base más poderosa, la cual a su vez tiene que llevar a cabo el reconocimiento de voz y posteriormente procesar el comando adecuadamente.

En la Figura 24 se muestra el protocolo de recolección de datos en acción. En la Figura 24a, el usuario inicia emitiendo un comando de voz al ambiente, uno o más nodos pueden detectar la presencia de audio; en este caso, como se muestra en la Figura 24b, los nodos A y B se percatan del hecho de que el usuario está emitiendo un comando de voz, cada uno de estos nodos envía un *mensaje de reporte de evento* (*ERM*, por sus siglas en inglés) al nodo líder (*CH*, por sus siglas en inglés), el cual, inmediatamente envía un *mensaje de selección de nodo* (*NSM*, por sus siglas en inglés) mediante broadcast a todo el cluster, este mensaje incluye la identidad del nodo que se seleccionó para capturar y transmitir las muestras de audio (Figura 24c); en la Figura 24d, el nodo seleccionado (B), entra en modo de *muestreo de alta frecuencia* (HFS) e inicia la captura de audio durante un período determinado; tras haber concluído este período, el nodo sale del modo *HFS* e inicia la transmisión de los *datos de audio* (AD)

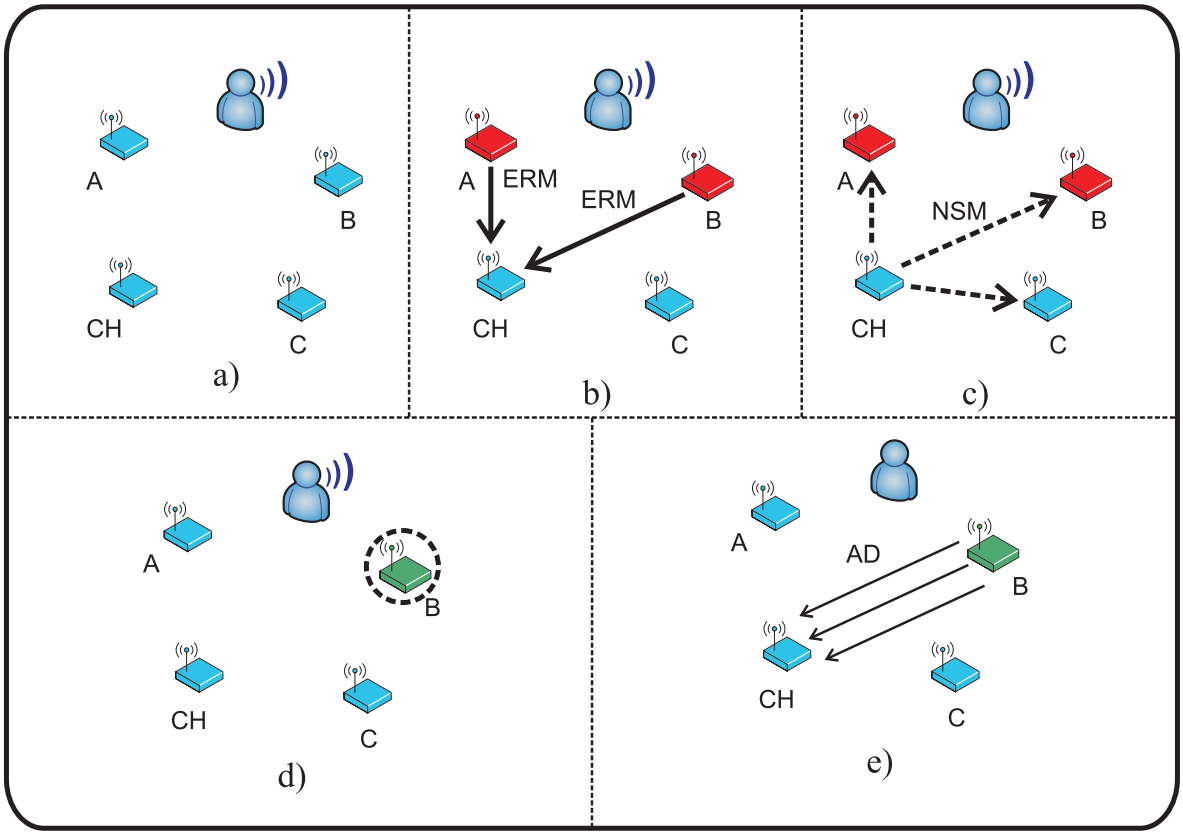

Figura 24. Esquema de captura de voz propuesto.

hacia el nodo líder (Figura 24e).

Funciones de seguridad

En algunos escenarios, la seguridad representa un requerimiento importante, por tal motivo, se integró un mecanismo para proveer seguridad contra ataques de reenvío y de paquetes de datos falsos. Al hacer esto, se evita que posibles intrusos utilicen paquetes previamente capturados por nodos comprometidos para duplicar comandos emitidos originalmente por fuentes autorizadas. Adicionalmente a esto, los reportes también son autentificados, es decir, solo los nodos que pertenecen al cluster pueden generar mensajes de reporte de eventos válidos. Para lograr esto, en el mensaje de reporte de evento se incluyó el valor encriptado de un contador que solamente lo conocen los nodos autorizados (miembros del cluster). Este contador se sincroniza en todos los nodos del cluster. Por lo tanto, si un intruso trata de falsificar un mensaje de reporte de evento, tiene que conocer el valor actual del contador así como la llave secreta utilizada para encriptarlo, de lo contrario, el nodo líder rechaza el mensaje de reporte de evento. Además, los nodos del cluster (incluyendo el nodo líder) mantienen rastro del estado en que se encuentran, por ejemplo: si el nodo líder no se encuentra en el estado de escuchar audio rechaza cualquier mensaje que contenga muestras de audio.

Al igual que en la aplicación presentada en la sección V.2.2, se seleccionó el algoritmo RC5-32/12/16 para generar los códigos MAC empleados en el protocolo. Cabe señalar que con este mismo algoritmo, se pudiesen encriptar los datos para proveer el servicio de confidencialidad en aplicaciones que así lo requieran.

Protocolo de sincronización del contador

Para garantizar el buen funcionamiento del protocolo propuesto, es indispensable que los nodos que forman parte del cluster estén sincronizados en todo momento. A pesar de que se utilizaron diferente tipo de nodos de sensado que en la aplicación presentada en la sección V.2.2, ambas utilizan un oscilador de cristal que presenta la mismas características. Por tal motivo, en esta aplicación se utiliza la misma técnica de sincronización del contador que en la aplicación antes mencionada.

Los mensajes de actualización del contador se deben encriptar por motivos de seguridad, esto debido a que si un intruso conoce el valor del contador, queda habilitado para generar mensajes falsos y de engañar al nodo líder para que éste acepte dichos mensajes.

Como se muestra en la instrucción 18, el nodo líder envía mediante broadcast un mensaje que contiene el nuevo valor del contador encriptado ($CNTR$) y el código MAC calculado con el encabezado del mensaje (HDR) y el nuevo valor del contador.

$$NodoLider \Rightarrow Nodos : E_{K_{cluster}}(CNTR) \| MAC(K_{cluster}, HDR \| CNTR) \qquad (18)$$

Esta técnica de sincronización del contador agrega un consumo de recursos adicionales en la red de sensores, pero se anticipa que estos mensajes de actualización del contador no se tienen que enviar con tanta frecuencia. La frecuencia de estos mensajes depende directamente de la longitud del período de sensado, así como también de la plataforma de implementación. En el capítulo de experimentos se abordan más detalles relacionados con la frecuencia de los mensajes de actualización del contador para la implementación prototipo que se presenta.

Figura 25. Hardware VR Stamp utilizado para reconocimiento de voz en la estación base.

Reconocimiento de voz

Para el reconocimiento de voz se utilizó el paquete de hardware *VR Stamp* de la compañía *Sensory Inc*[8] (mostrado en la Figura 25). El paquete *VR Stamp* es una herramienta que utiliza algoritmos basados en redes neuronales para el reconocimiento de patrones de voz. El *VR Stamp* consta de una tarjeta con un procesador de voz RSC-4128 a bordo, una memoria flash de 1 Mbit y una memoria EEPROM de 128 KB para datos. Esta tarjeta está conectada a la estación base (computadora personal) a través de un puerto USB.

En este capítulo se presentaron dos aplicaciones basadas en la propuesta del protocolo de manejo de información redundante en redes inalámbricas de sensores densamente pobladas. En cada una de las aplicaciones se presenta su diseño y aspectos específicos de la implementación prototipo utilizada para validar dicho la propuesta del protocolo. Posteriormente, en el siguiente capítulo, se describen los escenarios utilizados para el

[8]Sitio web: http://www.sensoryinc.com/products/, visitado el 23 de febrero del 2009.

desarrollo de experimentos de ambas aplicaciones desarrolladas, se muestran los resultados obtenidos y se discuten los mismos, con el fin de evaluar la eficiencia de las soluciones propuestas.

Capítulo VI

Prototipos y Experimentos

VI.1 Introducción

En este capítulo se abordan detalles de los prototipos que implementan las aplicaciones presentadas en el Capítulo V. Además se presentan los escenarios utilizados para el desarrollo de experimentos, así como los resultados más sobresalientes de dichos experimentos. Este capítulo se divide en dos secciones, en las cuales se presentan detalles y resultados de los experimentos de la aplicación de diseminación de datos y de la aplicación de recolección de voz respectivamente.

VI.2 Diseminación de datos de nodos miembro hacia nodo líder en redes jerárquicas de sensores.

Se llevaron a cabo una serie de experimentos al prototipo que se realizó en base a la propuesta presentada en la sección V.2, con el objetivo de cuantificar el consumo adicional de energía que introduce el protocolo propuesto. Se desarrollaron dos programas prototipo para una red de sensores compuesta por un solo cluster de 20 nodos, manejando un período de sensado de 5 segundos, con un ciclo de trabajo del 10% y transmitiendo a

0 dBm[1] (equivalente a 1 mW). El primer programa que se ejecutó implementa la simple técnica de "captura y envío", es decir, los nodos sensan los parámetros ambientales y los transmiten inmediatamente al nodo líder, se ejecutó dicho programa durante dos horas observando la cantidad de mensajes que se generan durante cada período de sensado. En el segundo programa, se implementa la propuesta presentada en la sección V.2, al igual que en el primer programa, el tiempo de observación fue de dos horas, y también se registraron la cantidad de mensajes generados por período de sensado, solo que en este caso, el programa se desarrolló de tal forma que los datos presenten redundancia desde 1 hasta el tamaño del cluster (20 en este caso).

Se comparó el protocolo propuesto en la sección V.2 contra el enfoque tradicional "captura y envío".

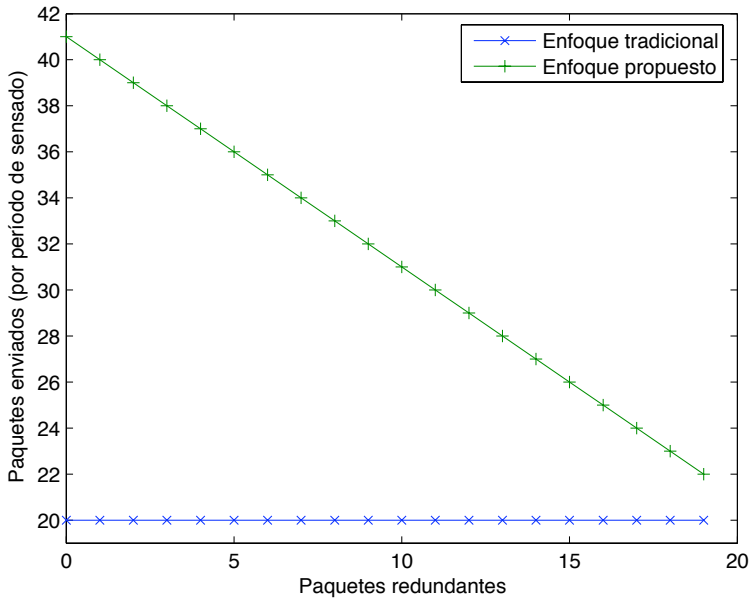

Figura 26. Número total de paquetes enviados vs. paquetes redundantes.

[1]El dBm se define como el nivel de potencia en decibeles en relación a un nivel de referencia de 1 mW.

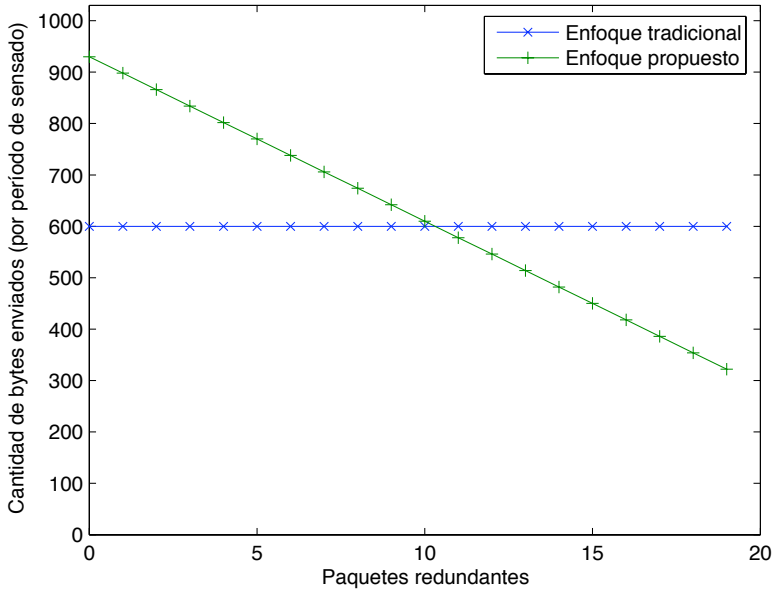

Figura 27. Tráfico generado (en bytes) vs. paquetes redundantes.

Para los experimentos de tráfico en el cluster de 20 nodos ya mencionado, se observó que la cantidad de mensajes transmitidos es siempre mayor en el protocolo propuesto que en el enfoque tradicional, esto se debe al procedimiento de dos fases introducido en la propuesta (ver Figura 26). Sin embargo, como se muestra en la Figura 27, en el protocolo propuesto se transmite una cantidad mucho menor de bytes; esto se debe al hecho de que los mensajes que transportan el código MAC son considerablemente más pequeños en comparación a los mensajes que transportan las lecturas. Por lo tanto, en este protocolo se introduce una cantidad total menor de tráfico que en el enfoque tradicional. En la Figura 26 y la Figura 27 se muestran las gráficas del número de mensajes enviados y el número de bytes enviados respectivamente contra el número de mensajes redundantes por período generados en el experimento del cluster de 20 nodos. Por ejemplo, si en un período determinado se producen 12 mensajes redundantes en el cluster, la cantidad total de tráfico es de 540 bytes en el protocolo, en comparación

a 600 bytes generados en el enfoque tradicional. Esto produce un ahorro de ancho de banda del 10%. Como se puede observar en la Figura 27, el ahorro de ancho de banda aumenta aún más conforme se incrementa el número de paquetes redundantes ya que la cantidad de tráfico introducido a la red es menor. Es importante indicar que el protocolo propuesto está orientado hacia aquellas aplicaciones que requieren de redes densamente pobladas, es decir, que tienen nodos ubicados uno muy cerca del otro. Por tal motivo, se espera que en este tipo de redes se genere un alto número de mensajes redundantes.

Debido a que el tiempo de vida de la red de sensores se anticipa que esté en el orden de meses. Se optó por utilizar cálculos analíticos en base a una simulación en Matlab, donde se procuró que las condiciones de la simulación fueran lo más apegadas a las del prototipo ya se mencionado. Considerando que se utilizaron motes *TMote Sky* en la implementación real, y dado el hecho de que estos nodos cuentan con un par de baterías AA como fuente de energía, se estimó el tiempo de vida para cada mote considerando diferentes niveles de redundancia de datos para el cluster. Se introdujeron las especificaciones técnicas de los motes a la simulación, particularmente, aquellas especificaciones relacionadas con el consumo de energía de los diferentes modos de operación de los motes.

Un par de baterías AA provee aproximadamente una corriente de 2.5 Ah, pero también se consideró que en la práctica es imposible el consumir la corriente en su totalidad debido a que después de cierto punto, el voltaje proporcionado por las baterías cae por debajo del umbral de operación requerido por los motes; sin embargo, es seguro suponer que se cuenta con una carga útil de 2200 mAh en las baterías (Mainwaring *et al.*, 2002).

Para las estimaciones obtenidas de consumo de energía, se observó que en los nodos

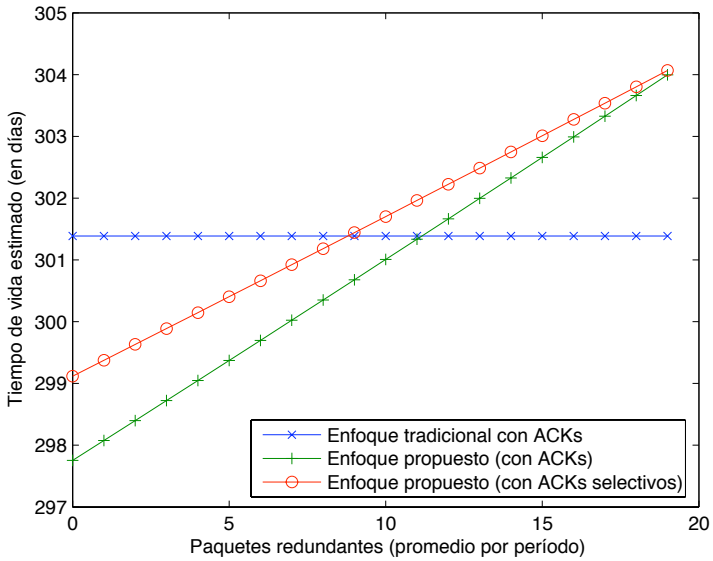

Figura 28. Tiempo de vida esperado en los nodos de sensado vs. número paquetes redundantes (en promedio) recibidos en el nodo líder utilizando señales ACK del protocolo de la capa MAC.

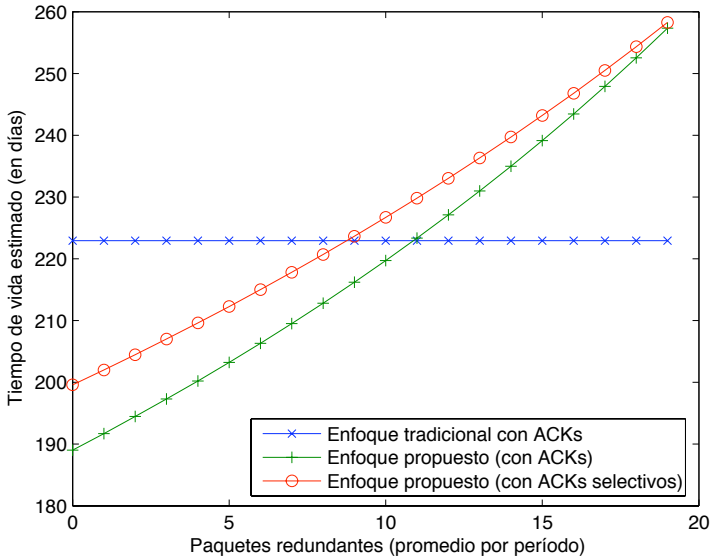

Figura 29. Tiempo de vida esperado en el nodo líder vs. número paquetes redundantes (en promedio) recibidos en el nodo líder utilizando señales ACK del protocolo de la capa MAC.

miembro el protocolo propuesto no introdujo un consumo adicional de energía, de hecho, en los casos donde existen altos niveles de redundancia se logró extender ligeramente el tiempo de vida de los nodos (cerca del 3% cuando mucho, ver Figura 28), esto se debe a que aunque en el protocolo propuesto se envían mensajes más cortos, se consume energía adicional al recibir las peticiones de datos del nodo líder; en esta plataforma se consume más energía al recibir que al transmitir. Por otra parte, en la Figura 29 se muestra que en el nodo líder se logró extender el tiempo de vida hasta en un 20% dependiendo de la redundancia. En esta gráfica, se muestra el enfoque tradicional junto con el protocolo propuesto en el cual se utilizan señales ACK en la capa MAC, también se muestra un tercer enfoque, en donde se utilizan señales ACK únicamente para el caso de los nodos a los cuales se les requieren sus datos (a este enfoque se le llamó *ACK selectivos*). Al utilizar este último, se mejoró considerablemente el desempeño de la propuesta en cuanto al consumo de energía se refiere; esto debido a que se ahorra energía al reducir el número de señales ACK enviadas del nodo líder hacia los nodos miembro del cluster.

En lo que se refiere a la frecuencia de los mensajes de actualización del contador para asegurar sincronización, se consideró que los motes *TMote Sky* utilizan un cristal de la compañía CITIZEN[2] con una tolerancia de ±20 ppm (pulsos por minuto) según las especificaciones técnicas. Por lo tanto, en el peor de los casos esto produce una desviación de 1 en cada 49,153 segundos de operación (≈ un segundo cada 13.5 horas en el peor de los casos). Si se considera un período de sensado de 5 segundos, el enviar un mensaje de actualización del contador cada 24 horas sería suficiente[3]. Por lo anterior, el costo en energía de agregar un mensaje diariamente es despreciable tanto

[2]Sitio web: http://www.citizencrystal.com, visitado el 3 de noviembre de 2008.

[3]Esto implica permitir un desvío máximo de 2 segundos.

en los nodos miembro como en el nodo líder.

VI.3 Diseminación de comandos de voz en una red inalámbrica de sensores.

Escenario de experimentación

Se instalaron dos clusters dentro de un apartamento pequeño como se muestra en la Figura 22 del Capítulo V, cada cluster está formado por 4 nodos (3 nodos de sensado y un nodo líder) y operaba con un ciclo de trabajo del 10% y un período de 1 segundo; cada cluster estaba a cargo de capturar los comandos de voz de un área especifica. Se utilizaron dos técnicas de captura diferentes en los experimentos; en la primera se utilizó una técnica de "captura y envío" en la que cada nodo que detecta la presencia de voz, graba tres segundos de audio, y al terminar de grabar, envían las muestras al nodo líder. La segunda técnica se basa en la propuesta descrita en la sección V.3: cuando un nodo detecta voz, envía un mensaje de notificación de evento al nodo líder, el cual a su vez, inmediatamente selecciona al nodo que está a cargo de grabar y enviar las muestras de audio. En ambos casos, cuando un nodo detecta audio, interrumpe su ciclo de encendido-apagado que se implementó originalmente para eficiencia energética, cuando un nodo concluye su transmisión, reanuda su ciclo.

Análisis de tráfico

En la Figura 30, se muestra el número de mensajes generados en la red cuando un usuario emite un comando de voz. Con la primer técnica se puede ver que a medida de que el número de nodos que se percatan de la existencia de voz en el ambiente,

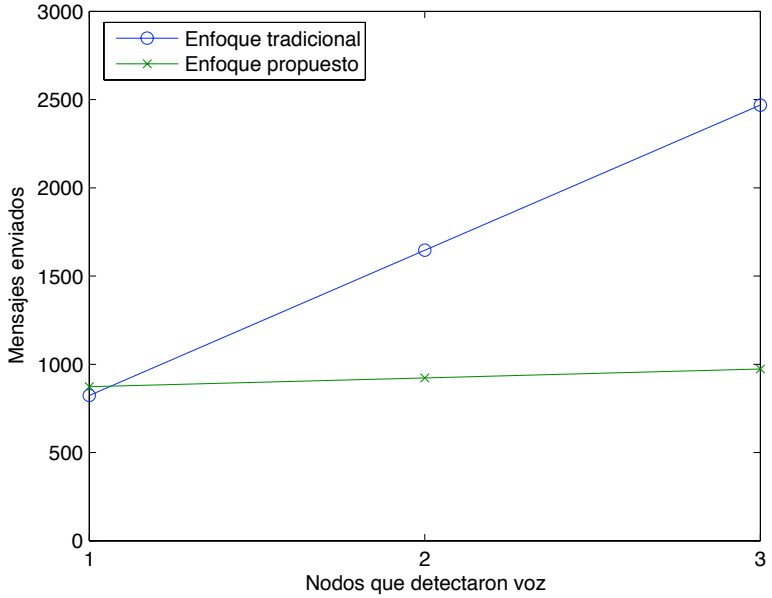

Figura 30. Gráfica del número de mensajes enviados vs. nodos que detectaron voz.

el número de mensajes aumenta linealmente. En contraste, en la segunda técnica, conforme aumenta el número de nodos que se percatan de la voz, el número de mensajes aumenta solo en uno por cada nodo adicional. Con este enfoque propuesto, solo se incrementa el número de mensajes en uno cada que un nodo más se percata de la voz, el mensaje adicional corresponde al mensaje de reporte de evento de ese nodo. Por lo tanto, se puede ver fácilmente que el enfoque propuesto es mucho más escalable que el primer enfoque.

Consumo de energía

Para los experimentos de consumo de energía, considerando que los nodos MicaZ utilizan un par de baterías, se estimó el tiempo de vida de cada nodo considerando que uno o más nodos pueden percatarse de la presencia de voz emitida por el usuario, por fines prácticos se consideran también un promedio de cien comandos de voz emitidos

diariamente. Se midió el consumo de energía en los nodos para los posibles escenarios en la implementación. Como ya se mencionó en los experimentos realizados de la sección anterior, se puede considerar que un par de baterías ofrece una carga utilizable de 2200 mAh.

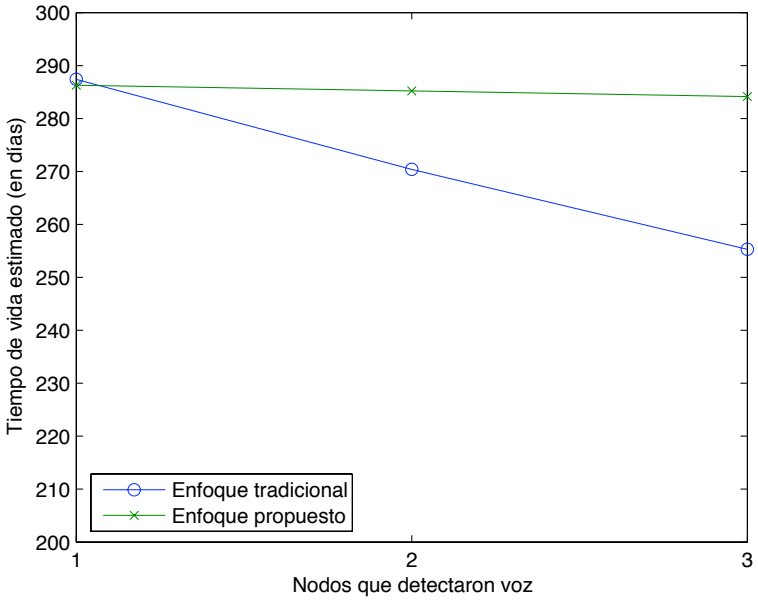

Figura 31. Tiempo de vida estimado en el nodo líder vs. número de nodos que detectaron voz.

En la Figura 31 se muestra que en el enfoque "captura y envío" el tiempo de vida estimado es muy dependiente del número de nodos del cluster que se percatan de la voz, en el peor de los casos (cuando tres nodos detectan la presencia de voz en cada período) el tiempo de vida se reduce en un 17%. En el protocolo propuesto, el tiempo de vida del cluster no depende de este número de nodos, esto es debido a que sin importar cuantos nodos se percaten de la presencia de voz, solamente se envía una copia del audio sensado. En ambos casos, para el nodo líder se consideró la cantidad de energía que consumen en estado activo, en estado de bajo consumo, la energía consumida al transmitir mensajes NSM (introducidos en el capítulo V), señales ACK y

la cantidad de energía utilizada al recibir mensajes ERM y muestras de audio. Como ya se ha mencionado, recibir datos consume mayor cantidad de energía que transmitir. En la Tabla II se muestra la cantidad de corriente que consume un mote MicaZ en los diferentes estados de operación.

Tabla II. Corriente consumida por los motes MicaZ.

Actividad	**Corriente consumida**
Estado de ocio (activo)	8.02mAh
Estado de bajo consumo (durmiendo)	16μAh
Transmitiendo	25.4 mAh
Recibiendo	27.7mAh

El tiempo de vida esperado[4] de los nodos miembros del cluster es muy dependiente del criterio de selección del nodo líder, como ya se mencionó, es muy recomendable que se utilice algún mecanismo de balanceo de cargas para prevenir que alguno de los nodos se agote en forma prematura. Por lo tanto, si se utiliza un esquema que balancee la carga uniformemente en todos los nodos del cluster se puede establecer que el tiempo de vida promedio de los nodos se multiplicaría n veces comparado con el enfoque tradicional, en este caso n sería el número de nodos de sensado en el cluster (sin incluir el nodo líder). Por ejemplo, considérese el cluster de tres nodos de sensado más el nodo líder, supóngase también que se emiten un promedio de 100 comandos diarios y que los tres nodos se percatan de la presencia de voz en cada uno de los comandos emitidos, si el nodo líder siempre solicita los datos al mismo nodo de sensado, el tiempo de vida

[4]Como el tiempo de vida esperado está en el orden de meses, no se han podido verificar con experimentos aún.

de éste es de aproximadamente de 142 días. En contraste, si el nodo líder utiliza un esquema simple "round-robin" alternando la selección de nodos, el tiempo de vida de cada nodo es de aproximadamente 426 días.

Reconocimiento de voz y pérdida de paquetes

En el enfoque propuesto hay un detalle que se puede considerar una desventaja: al tener que primero reportar la presencia de voz y al tener que esperar al nodo líder que envíe el comando de grabar antes de iniciar con la grabación se introduce un pequeño retardo que da como resultado que se pierda un pequeño número de muestras de audio (se midió un retardo de aproximadamente 65 ms). Sin embargo, esta pérdida es despreciable debido al hecho de que la plataforma VR Stamp que se mencionó anteriormente siguió siendo capaz de detectar el comando de voz.

Cabe señalar, que la estructura sintáctica de los comandos que se manejó en el prototipo estaba compuesta por tres elementos:

1. *Acción*. Lo que debe de llevar a cabo el comando.

2. *Sujeto*. El objeto sobre el cual se debe de llevar a cabo la acción.

3. *Información contextual*. Aquí se considera información adicional que indica de forma más específica al sistema de automatización, esto con el fin de eliminar ambigüedades. Entre esa información contextual se puede incluir: ubicación, estado actual o alguna otra característica propia que distinguen al elemento de los demás.

Algunos ejemplos de estos comandos son: "abrir-puerta-principal", "cerrar-llave-caliente", "encender-luz-estancia".

Una ventaja importante de la propuesta que no se había detectado inicialmente, es el hecho de que se elimina el problema de comandos duplicados desde la fuente, al contrario del enfoque "captura y envía" donde la estación base recibe copias múltiples del mismo comando emitido por el usuario. Debido a esto, se tiene que llevar a cabo procesamiento adicional en la estación base para poder eliminar la redundancia.

La plataforma VR Stamp fue capaz de reconocer cerca de el 94% de los comandos de voz emitidos aunque existió pérdida de paquetes. Se midió un promedio de 11% y 4% de pérdida de paquetes para el primero y el segundo enfoque respectivamente. Se supone que el enfoque "captura y envía" genera una mayor cantidad de paquetes perdidos debido a que hay mayor contención del canal dado de que más de un nodo requiere transferir volúmenes altos de datos hacia el nodo líder, mientras que en el protocolo propuesto solo el nodo seleccionado tiene que hacer esto.

Es importante señalar que pueden ocurrir colisiones; éstas son particularmente dañinas en el protocolo propuesto cuando se dan en la transmisión de los mensajes de reporte de eventos debido a que afectan directamente a la detección de redundancia. Sin embargo, el protocolo de capa MAC que se utiliza (B-MAC) reduce la pérdida de paquetes y aborda el problema de colisiones. Pero el utilizar un protocolo como B-MAC, aunado al hecho de que las WSN tienen un ancho de banda disponible bastante limitado (especialmente considerando aplicaciones de voz), introduce retrasos importantes en la comunicación. En este caso, se obtuvo un retraso promedio de 4.341 segundos, el cual se midió desde la conclusión del comando de voz emitido hasta el tiempo en que la totalidad del comando arribó a la estación base; esto puede parecer un retraso muy grande, pero es importante considerar que la aplicación prototipo está orientada hacia ambientes de cómputo ubicuo en el hogar, donde la respuesta en tiempo real no es un requerimiento primordial.

Frecuencia de actualización del contador

En lo que a la frecuencia de mensajes de actualización del contador se refiere, se consideró que los motes MicaZ utilizan el mismo oscilador de cristal que los motes TMote Sky empleados en la implementación del prototipo presentado en la sección VI.2. Además el formato del mensaje de actualización del contador es el mismo que se utilizó en dicho prototipo. Por lo tanto, se tomaron las mismas medidas para mantener la sincronía entre los miembros del cluster (enviar un mensaje de actualización diario), y como ya se mostró anteriormente, el introducir un mensaje adicional al día no incurre en un consumo considerable de energía.

Análisis de seguridad

Para poder determinar la eficiencia de los servicios de seguridad proveídos por el protocolo propuesto se analizó su comportamiento contra tres ataques comunes a la seguridad: inyección de tráfico, escuchar tráfico y reenvío de mensajes.

Inyección de Tráfico. El comportamiento observado en la red durante un ataque de inyección de tráfico varió dependiendo del tipo de mensajes insertados a la red. Si se inyecta un mensaje de actualización de contador, éste no produce ningún efecto adverso debido a que este mensaje se encuentra autentificado y se supone que el intruso no tiene acceso a las llaves criptográficas. Por otra parte, si se agrega un mensaje de reporte de evento (ERM) falso, el nodo líder lo rechaza debido a que también los ERM se autentifican utilizando el valor del contador encriptado. Si un intruso introduce datos de audio falsos, el nodo líder también rechaza estos mensajes si éste no ha recibido con anterioridad el ERM auténtico correspondiente; hay que recordar que el nodo líder lleva registro del estado actual en el que se encuentra el cluster.

Escuchar tráfico. Los ERM se encriptan utilizando el encriptador RC5, la seguridad del algoritmo recae en lo secreto de la llave. Sin embargo, los datos de audio no se encriptan. Se decidió no encriptarlos debido al costo asociado con hacerlo. Además, no se anticipó ningún problema potencial con que el intruso capture mensajes de audio de la red. Si él decidiera capturar el audio, lo podría hacer directamente desde el ambiente.

Reenvío de mensajes. Al utilizar un contador que cambia su valor períodicamente, se evita que el intruso utilice mensajes capturados previamente para engañar al nodo líder para que acepte mensajes ERM.

En este capítulo se presentaron los escenarios de experimentación utilizados sobre los dos prototipos implementados, se mostraron los resultados obtenidos tras la realización de los experimentos, así como una discusión acerca de dichos resultados. En el siguiente capítulo, se describen a grandes rasgos las aportaciones detectadas tras la realización del presente trabajo. Por otra parte, se fijan algunas directrices para trabajo futuro.

Capítulo VII

Conclusiones

Este trabajo de tesis se enfocó en limitar el efecto de la presencia de información redundante en Redes Inalámbricas de Sensores (WSN por sus siglas en inglés) donde se han colocado nodos en forma redundante con el fin de mejorar la confiabilidad global del sistema. En este tipo de escenarios frecuentemente se presentan redes densamente pobladas que se estructuran de manera jerárquica debido a que esta topología es más escalable y también puesto que resulta muy adecuada para aplicaciones donde muchas fuentes envían datos hacia una misma estación base o nodo sumidero (sink). Además, en este trabajo se presentan las implicaciones de trabajar con información redundante en WSN, lo cual puede traer como consecuencia el consumo excesivo de recursos de hardware y energía. Se hicieron dos aplicaciones que se basan en el manejo de información redundante mediante el protocolo propuesto con el fin de reducir el consumo de energía. Dichas aplicaciones se llevaron a la fase de implementación con el fin de realizar experimentos y validar si el uso del enfoque propuesto trae algún beneficio sobre los enfoques tradicionales de diseminación de datos de los nodos de sensado hacia el nodo líder del cluster.

VII.1 Logros

El principal logro alcanzado tras la realización de este trabajo es el de haber realizado una propuesta sobre la diseminación de datos redundantes en WSN. La eficiencia de dicha propuesta se validó posteriormente por medio de dos aplicaciones que emplean diseminación de datos desde los nodos de sensado hacia el nodo líder. Ambas implementaciones se basan en detectar la presencia de información redundante para obtener ahorros considerables en el consumo de energía, esto debido a la disminución en el número de operaciones de comunicación lograda.

En la primer aplicación se logró integrar un sistema de monitoreo ambiental orientado a escenarios de WSN redundantes; en la aplicación prototipo implementada se monitorean parámetros ambientales tales como temperatura, humedad y luminosidad. Tras la implementación y la realización de experimentos sobre dicho prototipo, se logró extender hasta en un 17% el tiempo de vida de la red, al reducir el número de operaciones de comunicaciones. Además, se utilizaron funciones MAC para la detección oportuna de información redundante, con lo cual se pudieron integrar funciones básicas de seguridad, pensando en posibles escenarios que pudieran demandar de servicios de seguridad. Otra contribución importante, es el hecho de que el integrar funciones de seguridad en dicho protocolo no incurrió en un consumo adicional de recursos, razón por la cual se puede ver esto como un servicio de valor agregado.

En la segunda aplicación implementada, se integró un sistema de captura de comandos de voz orientados a ambientes de cómputo ubicuo en el hogar. Una contribución importante de este esquema es el de haber utilizado una plataforma con fuertes limitaciones en recursos de hardware para la captura de audio. Además las WSN tradicionalmente se han utilizado para monitoreo de fenómenos en exteriores. Actualmente, en la

literatura existen pocas implementaciones basadas en WSN orientadas especificamente a ambientes de hogar, lo cual le da cierto valor al prototipo desarrollado. En forma similar al prototipo anterior, el diseño se basó en la premisa de que existe información redundante, en este caso se considera el hecho de que si se emite un comando de voz, dos o más nodos pudieran estar capturando el mismo comando. Además de esto, en los experimentos realizados a este segundo prototipo, se obtuvieron resultados prometedores. En el nodo líder particularmente, se extendió considerablemente el tiempo de vida en comparación al enfoque tradicional. Por otra parte, al igual que en la primera aplicación, se incluyeron servicios básicos de seguridad tales como la autentificación y confidencialidad sin consumir recursos adicionales, ya que las mismas funciones de seguridad se utilizan con el fin de detectar cuántos nodos han detectado la presencia de voz. Sin embargo, aunque uno de los objetivos de este prototipo fue el de enfocarlo hacia una aplicación para asistir a personas con algún impedimento físico, existen algunas limitaciones que tienen que abordarse posteriormente para poder llevarlo hacia el usuario final. Dichas limitaciones se describen en la siguiente sección.

Adicionalmente, aunque no fue parte del trabajo principal de la presente tesis, se exploró un enfoque bio-inspirado para la diseminación de datos en redes inalámbricas de sensores. En este sentido, se realizó una propuesta relacionada con la adaptación del algoritmo de infección, se presentaron resultados preliminares que muestran que el uso de enfoques alternativos como éste, trae beneficios considerables relacionados con el ahorro de recursos en comparación a enfoques tradicionales.

VII.2 Perspectivas y Trabajo Futuro

Después de haber concluido con este trabajo de tesis, aún existe mucho trabajo que se desprende de las propuestas y de las implementaciones de los prototipos realizados. Indudablemente, en la implementación del segundo prototipo existe aún más trabajo por hacer debido a que este prototipo se concibió con un enfoque más aplicativo:

- Una de las limitantes del segundo prototipo, radica en el hecho de que en los nodos no se realiza ningún tipo de reconocimiento de los comandos de voz. Esto debido a que los nodos tienen limitado poder de procesamiento. Actualmente, los nodos capturan cualquier señal de audio que detectan, lo cual potencialmente agotaría prematuramente su fuente de energía ya que la mayoría del audio capturado en un escenario real pudiese ser ruido ambiental. Para aminorar este efecto, se puede explorar la idea de incorporar algún prefijo que pudieran reconocer los nodos, de tal forma que una vez que identifiquen el prefijo, continúen con el proceso de captura y transmisión. Por ejemplo, se podría anteponer el prefijo "comando" antes del comando en sí.

- Los motes *MicaZ* utilizados en el segundo prototipo son de uso general. Si se diseña hardware específico para esta aplicación, eliminando los componentes que nos son útiles de los nodos MicaZ (sensores por ejemplo), y se integra algún microcontrolador ligeramente más poderoso; se puede mantener el mismo costo de producción de los nodos (o inclusive disminuir), mientras se tiene un hardware más adecuado con la aplicación, lo cual puede permitir el trasladar un mayor número de tareas a los nodos de la red. Por ejemplo, algo que ayuda mucho es el poder trasladar las tareas de reconocimiento de voz a los nodos de sensado, ya que al hacer esto, se reduce en gran cantidad las operaciones de comunicación, lo

cual extiende mucho más el tiempo de vida de la red.

- Desde el punto de vista aplicativo del prototipo implementado en la segunda aplicación también se desprende trabajo futuro. Primero que nada, es necesario hacer una evaluación exhaustiva de en qué medida la infraestructura de red de sensado puede ayudar a los usuarios potenciales, los cuales se anticipa que puedan ser personas de la tercera edad y/o con alguna discapacidad motriz.

- Posteriormente, también es necesario realizar un estudio de viabilidad tanto económica como física, esto último se refiere a determinar qué tan factible es integrar esta tecnología en las propias viviendas en las que habitan los usuarios potenciales.

- Sería útil también el analizar el comportamiento de la infraestructura de red, durante un período prolongado (de 4 a 8 meses por ejemplo).

- El protocolo propuesto, supone que ya se cuenta con un nodo líder predeterminado, ya que se pensó en aplicaciones para donde la ubicación física de cada uno de los nodos es controlada, es decir, los nodos se instalan uno a uno manualmente; resultaría interesante las implicaciones que tendría un escenario donde la ubicación no es conocida a priori, como en el caso donde los nodos se lanzan desde un avión o como se plantea en la idea conceptual del proyecto Smart Dust (mostrada en la Figura 1). Con escenarios dinámicos como los que se mencionan, primero que nada se requiere de un mecanismo de formación de clusters, así como de selección de nodo líder. Pero en lo concerniente al manejo de información redundante, quizás sea conveniente el diseñar un enfoque adaptativo; esto debido al hecho de que la densidad de los nodos en la red puede variar de un punto a otro en la misma red, ya que no se tiene control sobre la posición inicial que

ocupan los nodos en la red. Como se presentó en la sección de Experimentos y Resultados, para ciertos niveles de redundancia, el protocolo propuesto tiene un mejor desempeño en comparación a otros donde la redundancia es baja. Si se tiene una red donde la densidad de los nodo varía considerablemente de un punto a otro, en los puntos de baja densidad probablemente no se justifique el uso del protocolo propuesto mientras que en los puntos de alta densidad si sea atractivo utilizarlo. Por lo anterior, cada nodo líder tiene la responsabilidad de analizar el comportamiento de la red (determinar los niveles de redundancia de datos generados), con base en lo anterior, el nodo líder tiene que decidir si es conveniente utilizar el enfoque propuesto o utilizar el enfoque tradicional. Como resultado de lo anterior, se puede dar el caso en que de un cluster a otro se utilicen esquemas diferentes para la diseminación de datos, lo cual resulta interesante, ya que hasta donde se ha revisado en la literatura, no existen WSN con tales características.

- En los resultados presentados, se menciona que los ahorros más significativos se presentan en el nodo líder; dado que el nodo líder lleva a cabo más actividades que los nodos de sensado, su tiempo de vida es considerablemente menor (alrededor de un 20% menor). En la técnica propuesta, se considera que el nodo líder siempre es el mismo durante el tiempo de vida de la red, por lo cual el tiempo de vida del cluster es dependiente del tiempo de vida del nodo que fue designado como clúster. La integración de un mecanismo dinámico de elección y re-elección del nodo líder (rotación de nodo líder), funcionaría como un mecanismo de balanceo de carga, con lo cual existe la posibilidad de que se extienda el tiempo de vida del cluster en general. Ya existen mecanismos publicados para elección y re-elección de nodos líder que se basan en sistemas de votación mayoritaria (Brust *et al.*,

2007), en este caso los votantes son los nodos miembros del cluster. Resulta atractiva la adaptación de mecanismos como éste al protocolo propuesto en el presente trabajo. Para este fin, sería necesario por un lado evaluar los beneficios de integrar un mecanismo rotación de nodo líder, y por otro, evaluar si la cantidad de recursos requeridos no excede a la de dichos beneficios obtenidos.

En general, existen muchas posibilidades de integración de funcionalidades específicas que atiendan casos particulares. Sin embargo, como ya se ha visto en la literatura relacionada con WSN: no existe una solución única que atienda todas (o una gran mayoría) las aplicaciones. En el caso de WSN, el diseño de los protocolos de comunicación está intimamente relacionado con las particularidades de los escenarios de la aplicación para los cuales se van a utilizar. Por lo cual, lo que puede funcionar muy bien en un escenario, puede no ser tan funcional en otro.

REFERENCIAS

Akkaya, K. y Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, **3**(3): 325–349.

Akyildiz, I., Su, W., Sankarasubramaniam, Y., y Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, **40**(8): 102–114.

Allred, J., Hasan, A. B., Panichsakul, S., Pisano, W., Gray, P., Huang, J., Han, R., Lawrence, D., y Mohseni, K. (2007). Sensorflock: an airborne wireless sensor network of micro-air vehicles. En: *Proceedings of the 5th international conference on Embedded networked sensor systems*, páginas 117–129, Sydney, Australia. ACM. ISBN 978-1-59593-763-6.

Banerjee, S. y Khuller, S. (2001). A clustering scheme for hierarchical control in multi-hop wireless networks. En: *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, páginas 1028–1037, Anchorage, AK, USA. IEEE Press. ISBN 0-7803-7016-3.

Basha, E. A., Ravela, S., y Rus, D. (2008). Model-based monitoring for early warning flood detection. En: *Proceedings of the 6th International Conference on Embedded networked sensor systems*, páginas 295–308, Raleigh, NC, EE.UU. ACM. ISBN 978-1-59593-990-6.

Bellare, M., Kilian, J., y Rogaway, P. (2000). The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, **61**(3): 362–399.

Black, J. y Rogaway, P. (2005). CBC MACs for arbitrary-length messages: The three-key constructions. *Journal of Cryptology*, **18**(2): 111–131.

Boulis, A., Ganeriwal, S., y Srivastava, M. B. (2003). Aggregation in sensor networks: an energy-accuracy trade-off. *Ad Hoc Networks*, **1**(2-3): 317–331.

Britton, M., Shum, V., Sacks, L., y Haddadi, H. (2005). A biologically-inspired approach to designing wireless sensor networks. En: *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, páginas 256–266, Istanbul, Turquía. IEEE Press. ISBN 0-7803-8801-1.

Brust, M., Andronache, A., Rothkugel, S., y Benenson, Z. (2007). Topology-based clusterhead candidate selection in wireless ad-hoc and sensor networks. En: *Proceedings of the Second International Conference on Communication Systems Software and Middleware*, páginas 1–8, Bangalore, India. IEEE Press. ISBN 1-4244-0613-7.

Bulusu, N. y Jha, S. (2005). *Wireless sensor networks: a system perspective*. Artech House Publishers. ISBN 1-58053-867-3, páginas 1–326.

Burrell, J., Brooke, T., y Beckwith, R. (2004). Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, **3**(1): 38–45.

Carman, D. W., Kruus, P. S., y Matt, B. J. (2000). Constraints and approaches for distributed sensor network security. Reporte técnico 00-010, NAI Labs, The Security Research Division.

Çam, H., Özdemir, S., Nair, P., Muthuavinashiappan, D., y Sanli, H. O. (2006). Energy-efficient secure pattern based data aggregation for wireless sensor networks. *Computer Communications*, **29**(4): 446–455.

Chan, H., Perrig, A., y Song, D. X. (2003). Random key predistribution schemes for sensor networks. En: *Proceedings of the IEEE Symposium on Security and Privacy*, páginas 197–213, Berkeley, CA, EE.UU. IEEE Computer Society. ISBN 0-7695-1940-7.

Chu, M., Haussecker, H., Zhao, F., Chu, M., Haussecker, H., y Zhao, F. (2002). Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, **16**(3): 293–313.

Daemen, J. y Rijmen, V. (2000). Rijndael for AES. En: *AES Candidate Conference*, páginas 343–348, New York, NY, EE.UU.

Demers, A., Gehrke, J., Rajaraman, R., Trigoni, N., y Yao, Y. (2003). The cougar project: a work-in-progress report. *SIGMOD Rec.*, **32**(4): 53–59.

Douceur, J. R. (2002). The sybil attack. En: P. Druschel, M. F. Kaashoek, y A. I. T. Rowstron, editores, *IPTPS*, Vol. 2429 de *Lecture Notes in Computer Science*, páginas 251–260. Springer. ISBN 3-540-44179-4.

Dressler, F., Krger, B., Fuchs, G., y German, R. (2005). Self-organization in sensor networks using bio-inspired mechanisms. En: *Proceedings of 18th ACM/GI/ITG International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing (ARCS'05): Workshop Self-Organization and Emergence*, páginas 139–144.

Eschenauer, L. y Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. En: V. Atluri, editor, *Proceedings of the ACM Conference on Computer and Communications Security*, páginas 41–47. ACM. ISBN 1-58113-612-9.

Flavin, C. (1991). Understanding fault-tolerant distributed systems. *Commun. ACM*, **34**(2): 56–78.

Gandhi, S., Suri, S., y Welzl, E. (2007). Catching elephants with mice: sparse sampling for monitoring sensor networks. En: *Proceedings of the 5th International Conference*

on Embedded networked sensor systems, páginas 261–274, Sydney, Australia. ACM. ISBN 978-1-59593-763-6.

Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., y Wicker, S. (2002). An empirical study of epidemic algorithms in large scale multihop wireless networks. Reporte técnico, Intel Research Labs @ Berkeley.

García-Macías, J. A. y Gómez, J. (2007). MANET versus WSN. En: N. P. Mahalik, editor, *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*, páginas 369–388. Springer-Verlag. ISBN 3540373640.

Gu, L., Jia, D., Vicaire, P., Yan, T., Luo, L., Tirumala, A., Cao, Q., He, T., Stankovic, J. A., Abdelzaher, T., y Krogh, B. H. (2005). Lightweight detection and classification for wireless sensor networks in realistic environments. En: *Proceedings of the 3rd International Conference on Embedded networked sensor systems*, páginas 205–217, San Diego, CA, EE.UU. ACM. ISBN 1-59593-054-X.

Hartung, C., Balasalle, J., y Han, R. (2005). Node compromise in sensor networks: The need for secure systems. Reporte técnico CU-CS-990-05, Department of Computer Science University of Colorado at Boulder.

Heinzelman, W. B., Chandrakasan, A. P., y Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, **1**(4): 660– 670.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E., y Pister, K. S. J. (2000). System architecture directions for networked sensors. En: *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, páginas 93–104, Cambridge, MA, EE.UU. ACM. ISBN 1-58113-317-0.

Hill, J., Horton, M., Kling, R., y Krishnamurthy, L. (2004). The platforms enabling wireless sensor networks. *Communications of the ACM*, **47**(6): 41–46.

Hoblos, G., Staroswiecki, M., y Aitouche, A. (2000). Optimal design of fault tolerant sensor networks. En: *Proceedings of the IEEE International Conference on Control Applications*, páginas 467–472, Anchorage, AK, EE.UU. IEEE Press. ISBN 0-7803-6562-3.

Hu, L. y Evans, D. (2003). Secure aggregation for wireless network. En: *Proceedings of the Symposium on Applications and the Internet Workshops*, páginas 384–394, Orlando, FL, EE.UU. IEEE Computer Society. ISBN 0-7695-1873-7.

Hu, Y.-C., Perrig, A., y Johnson, D. B. (2002). Wormhole detection in wireless ad hoc networks. Reporte técnico TR01-384, Department of Computer Science, Rice University.

Hwang, J. y Kim, Y. (2004). Revisiting random key pre-distribution schemes for wireless sensor networks. En: *Proceedings of the 2nd Workshop on Security of ad hoc and sensor networks*, páginas 43–52, Washington, DC, EE.UU. ACM. ISBN 1-58113-972-1.

IEEE (2003). *802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, página ????

Intanagonwiwat, C., Govindan, R., y Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. En: *Proceedings of the 6th annual International Conference on Mobile computing and networking*, páginas 56–67, Boston, MA, EE.UU. ACM. ISBN 1-58113-197-6.

Intille, S. S. (2002). Designing a home of the future. *IEEE Pervasive Computing*, **1**(2): 76–82.

ISO/IEC 9797 (1989). *Data cryptographic techniques - Data integrity mechanism using a cryptographic check function employing a block cipher algorithm*. International Standards Organization (ISO).

Kafri, R., Springer, M., y Pilpel, Y. (2009). Genetic redundancy: New tricks for old genes. *Cell Magazine*, **136**(3): 389–392.

Kahn, J. M., Katz, R. H., y Pister, K. S. J. (1999). Next century challenges: mobile networking for "smartdust". En: *Proceedings of the 5th International Conference on Mobile computing and networking*, páginas 271–278, Seattle, WA, EE.UU. ACM. ISBN 1-58113-142-9.

Kaiser, W. J., Pottie, G. J., Srivastava, M., Sukhatme, G. S., nor, J. V., y Estrin, D. (2003). Networked infomechanical systems (nims) for ambient intelligence. Reporte técnico.

Kansal, A., Potter, D., y Srivastava, M. B. (2004). Performance aware tasking for environmentally powered sensor networks. En: *Proceedings of the Joint International Conference on Measurement and modeling of computer systems*, páginas 223–234, New York, NY, EE.UU. ACM. ISBN 1-58113-873-3.

Karlof, C. y Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, **1**(2-3): 293–315.

Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E. D., Starner, T., y Newstetter, W. (1999). The aware home: A living laboratory for ubiquitous computing research. En: *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization,*

and Architecture, páginas 191–198, Londres, Reino Unido. Springer-Verlag. ISBN 3-540-66596-X.

Kim, N., Heo, J., Kim, H. S., y Kwon, W. H. (2008). Reconfiguration of clusterheads for load balancing in wireless sensor networks. *Computer Communications*, **31**(1): 153 – 159.

Klues, K., Hackmann, G., Chipara, O., y Lu, C. (2007). A component-based architecture for power-efficient media access control in wireless sensor networks. En: *Proceedings of the 5th International Conference on Embedded networked sensor systems*, páginas 59–72, Sydney, Australia. ACM. ISBN 978-1-59593-763-6.

Lees, J. M., Johnson, J. B., Ruiz, M., Troncoso, L., y Welsh, M. (2008). Reventador volcano 2005: Eruptive activity inferred from seismo-acoustic observation. *Journal of Volcanology and Geothermal Research*, **176**(1): 179 – 190. Recent and active volcanism in the Ecuadorian Andes.

Liao, D. y Sarabandi, K. (2006). Network of rf ground sensors for applications in precision agriculture. En: *Proceedings of the International Conference on Geoscience and Remote Sensing Symposium*, páginas 3943–3946, Denver, CO, EE.UU. IEEE Press. ISBN 0-7803-9510-7.

Liu, D., Ning, P., y Li, R. (2005). Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, **8**(1): 41–77.

Luo, L., Cao, Q., Huang, C., Abdelzaher, T., Stankovic, J. A., y Ward, M. (2007). Enviromic: Towards cooperative storage and retrieval in audio sensor networks. En: *Proceedings of the 27th International Conference on Distributed Computing Systems*, página 34, Toronto, Canad. IEEE Computer Society. ISBN 0-7695-2837-3.

Madden, S. y Hellerstein, J. M. (2002). Distributing queries over low-power wireless sensor networks. En: *Proceedings of the International Conference on Management of data*, páginas 622–622, Madison, WI, EE.UU. ACM. ISBN 1-58113-497-5.

Madden, S., Franklin, M. J., Hellerstein, J. M., y Hong, W. (2002a). Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, **36**(SI): 131–146.

Madden, S., Szewczyk, R., Franklin, M. J., y Culler, D. (2002b). Supporting aggregate queries over ad-hoc wireless sensor networks. En: *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications*, página 49, Washington, DC, USA. IEEE Computer Society. ISBN 0-7695-1647-5.

Mainwaring, A. M., Culler, D. E., Polastre, J., Szewczyk, R., y Anderson, J. (2002). Wireless sensor networks for habitat monitoring. En: C. S. Raghavendra y K. M. Sivalingam, editores, *WSNA*, páginas 88–97. ACM. ISBN 1-58113-589-0.

Martínez, K., Hart, J. K., y Ong, R. (2009). Deploying a wireless sensor network in iceland. En: *Proceedings of the 3rd International Conference on GeoSensor Networks*, páginas 131–137, Oxford, Reino Unido. Springer-Verlag. ISBN 978-3-642-02902-8.

Mozer, M. C. (1998). The neural network house: An environment that adapts to its inhabitants. En: *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, páginas 110–114, Menlo Park, CA, EE.UU. AAAI Press. ISBN 978-1-57735-418-5.

National Institute of Standards and Technology (1995). *FIPS PUB 180-1: Secure Hash Standard*. National Institute for Standards and Technology, Gaithersburg, MD, USA. Supersedes FIPS PUB 180 1993 May 11.

National Institute of Standards and Technology (1999). *FIPS PUB 46-3: Data Encryption Standard (DES)*. National Institute for Standards and Technology, Gaithersburg, MD, USA. supersedes FIPS 46-2.

Olague, G., de Vega, F. F., Pérez, C. B., y Lutton, E. (2004). The infection algorithm: An artificial epidemic approach for dense stereo matching. En: X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. M. Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, y H.-P. Schwefel, editores, *PPSN*, Vol. 3242 de *Lecture Notes in Computer Science*, páginas 622–632. Springer. ISBN 3-540-23092-0.

Palafox, L. E. y García-Macías, J. A. (2006). A bio-inspired approach for data dissemination in wireless sensor networks. *INFOCOMP Journal of Computer Science*, **5**(3): 19–27.

Palafox, L. E. y García-Macías, J. A. (2008). Secure data recollection for redundantly deployed wireless sensor networks. En: *Proceedings of the 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, páginas 1–6, Newport Beach, CA, EE.UU. IEEE Press. ISBN 978-1-4244-2099-5.

Palafox, L. E. y García-Macías, J. A. (2009). Deploying a voice capture sensor network system for a secure ubiquitous home environment. *International Journal of Communication Systems*, **22**(9): 1199–1212.

Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., y Culler, D. E. (2002). SPINS: Security protocols for sensor networks. *Wireless Networks*, **8**(5): 521–534.

Perrig, A., Stankovic, J. A., y Wagner, D. (2004). Security in wireless sensor networks. *Communications of the ACM*, **47**(6): 53–57.

Pottie, G. J. y Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, **43**(5): 51–58.

Przydatek, B., Song, D. X., y Perrig, A. (2003). SIA: secure information aggregation in sensor networks. En: *Proceedings of the 1st International Conference on Embedded networked sensor systems*, páginas 255–265, Los Angeles, CA, EE.UU. ACM. ISBN 1-58113-707-9.

Rivest, R. L. (1994). The RC5 Encryption Algorithm. En: B. Preneel, editor, *Fast Software Encryption*, Vol. 1008 de *Lecture Notes in Computer Science*, páginas 86–96. Springer.

Römer, K. y Mattern, F. (2004). The design space of wireless sensor networks. *Wireless Communications*.

Roundy, S., Wright, P. K., y Rabaey, J. M. (2004). *Energy Scavenging for Wireless Sensor Networks: With Special Focus on Vibrations*. Kluwer Academic Publishers, Norwell, MA, USA. ISBN 1402076630.

Schneier, B. (1996). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley, segunda edición.

Shah, R. C., Roy, S., Jain, S., y Brunette, W. (2003). Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, **1**(2-3): 215 – 233. Sensor Network Protocols and Applications.

Shen, C., Srisathapornphat, C., y Jaikaeo, C. (2001). Sensor information networking architecture and applications. *IEEE Personal Communications Magazine*, **8**(4): 52–59.

Shih, E., Cho, S.-H., Ickes, N., Min, R., Sinha, A., Wang, A., y Chandrakasan, A. (2001). Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. En: *Proceedings of the 7th annual international conference on Mobile computing and networking*, páginas 272–287, New York, NY, USA. ACM. ISBN 1-58113-422-3.

Sibley, G., Rahimi, M., y Sukhatme, G. (2002). Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks. En: *Proceedings of the IEEE International Conference on Robotics and Automation*, páginas 1119–1124, Washington, DC, EE.UU. IEEE Press. ISBN 0-7803-7272-7.

Simon, G., Maróti, M., Lédeczi, A., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., y Frampton, K. (2004). Sensor network-based countersniper system. En: *Proceedings of the 2nd international conference on Embedded networked sensor systems*, páginas 1–12, Baltimore, MD, EE.UU. ACM. ISBN 1-58113-879-2.

Stankovic, J. A., Abdelzaher, T. F., Lu, C., Sha, L., y Hou, J. C. (2003). Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, **91**(7): 1002–1022.

Su, H. y Zhang, X. (2006). Energy-efficient clustering system model and reconfiguration schemes for wireless sensor networks. En: *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, páginas 99–104, Princeton, NJ, EE.UU. IEEE Press. ISBN 1-4244-0349-9.

Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., y Estrin, D. (2004). Habitat monitoring with sensor networks. *Communications of the ACM*, **47**(6): 34–40.

Tennenhouse, D. (2000). Proactive computing. *Communications of the ACM*, **43**(5): 43–50.

Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., y Hong, W. (2005). A macroscope in the redwoods. En: *Proceedings of the 3rd International Conference on Embedded networked sensor systems*, páginas 51–63, San Diego, CA, EE.UU. ACM. ISBN 1-59593-054-X.

van Dam, T. y Langendoen, K. (2003). An adaptive energy-efficient mac protocol for wireless sensor networks. En: *Proceedings of the 1st International Conference on Embedded networked sensor systems*, páginas 171–180, Los Angeles, CA, EE.UU. ACM. ISBN 1-58113-707-9.

Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., y Corke, P. (2005). Data collection, storage, and retrieval with an underwater sensor network. En: *Proceedings of the 3rd International Conference on Embedded networked sensor systems*, páginas 154–165, San Diego, CA, EE.UU. ACM. ISBN 1-59593-054-X.

Wang, X., Gu, W., Chellappan, S., Xuan, D., y Lai, T. H. (2005a). Sacrificial node-assisted defense against search-based physical attacks in sensor networks. Reporte técnico, Department of Computer Science and Engineering, Ohio State University.

Wang, X., Gu, W., Schosek, K., Chellappan, S., y Xuan, D. (2005b). Sensor network configuration under physical attacks. En: X. Lu y W. Zhao, editores, *ICCNMC*, Vol. 3619 de *Lecture Notes in Computer Science*, páginas 23–32. Springer. ISBN 3-540-28102-9.

Warneke, B., Last, M., Liebowitz, B., y Pister, K. S. J. (2001). Smart dust: Communicating with a cubic-millimeter computer. *Computer*, **34**(1): 44–51.

Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., y Nagpal, R. (2005). Firefly-inspired sensor network synchronicity with realistic radio effects. En: *Proceedings of the 3rd International Conference on Embedded networked sensor systems*, páginas 622–622, San Diego, CA, EE.UU. ACM. ISBN 1-58113-497-5.

Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J., y Welsh, M. (2006). Fidelity and yield in a volcano monitoring sensor network. En: *Proceedings of the 7th Symposium*

on Operating systems design and implementation, páginas 381–396, Seattle, WA, EE.UU. USENIX Association. ISBN 1-931971-47-1.

Woo, A. y Culler, D. E. (2001). A transmission control scheme for media access in sensor networks. En: *Proceedings of the 7th annual international conference on Mobile computing and networking*, páginas 221–235, Roma, Italia. ACM. ISBN 1-58113-422-3.

Yao, Y. y Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, **31**(3): 9–18.

Yao, Y. y Gehrke, J. (2003). Query processing in sensor networks. En: *Proceedings of the First Biennial Conference on Innovative Data Systems Research*, páginas 1–12, Asilomar, CA, EE.UU.

Zhao, F., Shin, J., y J.Reich (2002). Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, **19**(2): 61–7.

Apéndice A

Seguridad en Redes Inalámbricas de Sensores

A.1 Introducción

Las redes inalámbricas de sensores han cobrado popularidad recientemente, esto debido principalmente al hecho de que ofrecen una solución alternativa de bajo costo a una gran variedad de problemas reales (Akyildiz *et al.*, 2002). Su bajo costo hace posible la implantación de grandes arreglos de sensores en una gran diversidad de condiciones, las cuales en la mayoría de los casos resultan desfavorables en comparación a ambientes en los que se instalan las redes tradicionales. Las redes de sensores como ya se ha mencionado anteriormente en este trabajo, introducen extremas limitaciones de recursos, esto principalmente debido a la falta de espacio de almacenamiento y energía. Ambas limitaciones representan grandes obstáculos en la integración de técnicas de seguridad tradicionales en redes inalámbricas de sensores. Los canales de comunicaciones poco confiables que se utilizan en este tipo de redes y el hecho de que en su mayoría operan de forma desatendida hacen aún más difícil el integrar medidas de seguridad a ellas. Como se menciona en (Perrig *et al.*, 2002), las redes inalámbricas de sensores comúnmente ofrecen las características de procesamiento de computadoras de hace ya algunas décadas. Además de esto, la tendencia de la industria es la de reducir el costo de los sensores inalámbricos manteniendo un mismo poder de cómputo. Basándose en esa idea, algunos investigadores han empezado a enfrentar el reto de maximizar la

capacidad de procesamiento y reducir el consumo de energía de las redes de sensores protegiéndolas a la vez contra posibles ataques.

Algunos aspectos de las redes inalámbricas de sensores ya han sido abordados entre ellos ruteo seguro y eficiente, agregación de datos, integración de grupos y otros más.

Además de esos problemas tradicionales de seguridad, se observa que muchas técnicas de redes de sensores de propósito general suponen un ambiente confiable, lo cual no es el caso para muchas de las aplicaciones reales de redes de sensores, que requieren de cierta confiabilidad para operar adecuadamente. Por tal motivo, algunos investigadores han estado trabajando en construir un modelo de confianza entre nodos para resolver aquellos problemas que van más allá de la seguridad criptográfica. Por otra parte, existen varios ataques diseñados para explotar los canales de comunicación poco confiables en los cuales operan las redes de sensores. Además, debido al modo de operación desatendido que presentan las redes de sensores, los ataques físicos a los nodos juegan un papel importante en la operación de este tipo de redes. Debido a esto, aunque no es la idea central del presente trabajo, en este capítulo se presenta un compendio de ataques físicos y sus respectivos mecanismos de defensa, lo cual típicamente es ignorado en la mayoría de los trabajos de investigación en el el área de seguridad en redes de sensores.

En este apéndice se presentará además un resumen sobre los obstáculos existentes para la seguridad en redes inalámbricas de sensores, los requerimientos de seguridad de una red de sensores. Una clasificación de los ataques más relevantes a los que están expuestas este tipo de redes y finalmente las medidas de seguridad que han sido propuestas en la literatura para contrarrestarlos.

A.2 Obstáculos de la seguridad en redes inalámbricas de sensores

Para desarrollar mecanismos útiles de seguridad es necesario considerar las limitaciones que presentan las WSN y que ya han sido mencionadas anteriormente (Carman *et al.*, 2000).

A.2.1 Recursos extremadamente limitados

Todos los mecanismos de seguridad requieren de cierta cantidad de recursos para su implementación, incluyendo memoria para datos, para código, y energía para alimentar el nodo. Sin embargo, estos recursos están muy limitados en los nodos que forman parte de las WSN.

- **Limitantes de memoria**. Un sensor es un dispositivo con una pequeña cantidad de memoria y poco espacio de almacenamiento para código. Para poder implementar un mecanismo de seguridad eficiente es necesario limitar el tamaño del código de los algoritmos que se utilicen. Por ejemplo, un sensor típico cuenta con un microcontrolador de 8 bits que corre a 4 MHz y que solo posee un total 8 KB de memoria. Considerando esa limitación, el software implementado para el sensor también debe de ser bastante pequeño. El espacio utilizado por TinyOS (el sistema operativo más utilizado en redes de sensores) es de 4KB (Hill *et al.*, 2000), mientras que el calendarizador de tareas ocupa solo 178 bytes. Por lo tanto, el tamaño del código para todas las funciones de seguridad también debe de ser muy pequeño.

- **Limitante de energía**. La energía es la limitante más grande en las redes de

sensores. En la mayoría de los casos se supone que una vez que se instalan los nodos de la red de sensores, no pueden ser reemplazados fácilmente ni tampoco recargados. Por lo tanto, la carga de la batería que lleven consigo en el momento de ser instalados debe de ser conservada para extender la vida de cada nodo y con ello, de la red de sensores en su totalidad. Cuando se implementa una función de seguridad o un protocolo en un nodo sensor, el impacto de la energía consumida al agregar la función de seguridad debe de ser considerado. Cuando se agrega seguridad a un nodo sensor, debe de ser de principal interés el impacto que la seguridad tendría en el tiempo de vida del nodo. La energía extra consumida por los nodos sensores debido a la seguridad se relaciona con el procesamiento requerido por las funciones de seguridad (por ejemplo: encriptación, decriptación, firma de datos, verificación de firmas), la energía requerida para transmitir datos relacionados con la seguridad o sobrecosto (por ejemplo: vectores de inicialización requeridos para encriptar/desencriptar) y la energía requerida para almacenar parámetros de seguridad de manera segura (por ejemplo: almacenamiento de llaves criptográficas).

A.2.2 Comunicación poco confiable

La comunicación poco confiable que ofrece el medio inalámbrico es otro obstáculo para la seguridad en las redes de sensores. La seguridad de la red depende enormemente de un protocolo definido, el cual a su vez depende de la comunicación de la red.

- **Transferencias no confiables**. Normalmente el ruteo basado en paquetes de una red de sensores no es orientado a conexión, lo cual hace a la red inherentemente no confiable. Los paquetes se pueden dañar debido a errores en el canal

de comunicación o incluso pueden ser descartados por aquellos nodos que se encuentren congestionados, lo cual conlleva a paquetes perdidos. El hecho de que se presente una tasa de error alta obliga al desarrollador de software a reservar recursos para el manejo de errores. Lo más importante aún, es el hecho de que si un protocolo no cuenta con los mecanismos adecuados para el manejo de errores es posible que se pierdan paquetes críticos de seguridad. Estos pueden incluir por ejemplo: una llave criptográfica.

- **Conflictos**. Aunque el canal de comunicaciones fuera confiable, la comunicación aún podría ser no confiable. Esto debido a la naturaleza tipo broadcast que presenta en las redes de sensores. Si existen colisiones en medio de una transferencia, ocurrirían conflictos y la transferencia misma fallaría. En una red de sensores densamente poblada, esto puede ser un gran problema. En (Akyildiz *et al.*, 2002) se pueden consultar más detalles acerca de los efectos de la comunicación inalámbrica.

- **Latencia**. El ruteo multi-saltos, la congestión de la red y el procesamiento realizado por los nodos pueden introducir latencia en la red, dificultando así la sincronización entre los nodos de la red. Los problemas de sincronización pueden resultar críticos para la seguridad de la red donde los mecanismos de seguridad dependen de reporte de eventos y distribución de llaves criptográficas. En (Stankovic *et al.*, 2003) se presentan algunos detalles sobre comunicación en tiempo real en redes inalámbricas de sensores.

A.2.3 Operación desatendida

En la mayoría de las aplicaciones de redes inalámbricas de sensores, los nodos se dejan desatendidos por largos periodos de tiempo. Enseguida se mencionan tres principales inconvenientes de dejar desatendida la red de sensores:

- **Exposición a ataques fisicos**. La red puede ser instalada en un ambiente abierto a adversarios, condiciones climatológicas adversas, etc. Por lo tanto, la probabilidad de que un nodo sensor sufra un ataque físico en tal ambiente es mucho más alta que el de las computadoras típicas, las cuales están ubicadas en un lugar seguro y únicamente se enfrentan a ataques por medio de la red.

- **Administradas remotamente**. La administración remota de una red de sensores hace prácticamente imposible el detectar ataques físicos ni problemas de mantenimiento de la red (por ejemplo: bajo nivel de carga en la batería. Quizás el caso más extremo de esto es cuando un nodo sensor utilizado en una aplicación militar de reconocimiento de terreno: en tal caso, el nodo ya no tendría contacto físico con el usuario una vez instalado.

- **No existe punto central de administración**. Una red de sensores debe de ser una red distribuida sin un punto central de administración. Esto aumentaría la vida de la red. Sin embargo, si el diseño no es adecuado, la organización de la red resultaría difícil, ineficiente y frágil.

Concretamente, mientras más tiempo permanezca desatendida la red de sensores, más probable resultaría que un adversario comprometiera a uno de sus nodos.

A.3 Requerimientos de seguridad

Una red inalámbrica de sensores cuenta con algunas similitudes con las redes tradicionales de computadoras, pero también presentan requerimientos propios que son exclusivos a éstas. Por tal motivo, se puede pensar que los requerimientos de seguridad de las redes inalámbricas de sensores, conjuntan tanto los requerimientos de seguridad de las redes tradicionales como también los requerimientos únicos de las redes inalámbricas de sensores.

A.3.1 Confidencialidad de datos

La confidencialidad de datos es el problema más importante de seguridad en redes. Cada red con cualquier enfoque de seguridad probablemente aborde este problema primero que ningún otro. En redes de sensores la confidencialidad se relaciona con lo siguiente (Carman *et al.*, 2000; Perrig *et al.*, 2002):

- Una red de sensores no debe filtrar lecturas de los sensores a sus vecinos. Especialmente en aplicaciones militares donde los datos almacenados en un nodo sensor pueden ser altamente confidenciales.

- En muchas aplicaciones los nodos comunican datos altamente confidenciales (por ejemplo: distribución de llaves), de tal manera que es muy importante el construir un canal seguro en una red inalámbrica de sensores.

- La información pública de los sensores, tales como su identidad y sus llaves públicas también deben de ser encriptadas hasta cierto punto para protegerse de ataques de análisis de tráfico.

El enfoque tradicional para mantener en secreto información confidencial es el de encriptar los datos con una llave secreta que solo el receptor conoce, obteniendo así la confidencialidad.

A.3.2 Integridad de los datos

Con la implementación de la confidencialidad un adversario puede quedar deshabilitado de robar información de la red de sensores. Sin embargo, esto no significa que los datos están seguros. El adversario puede modificar los datos al grado de afectar completamente el funcionamiento de la red. Por ejemplo, un nodo malicioso puede agregar o quitar ciertos fragmentos a un paquete. Luego este paquete puede ser enviado al receptor original. La pérdida o la corrupción de los datos puede ocurrir inclusive sin la presencia de un nodo malicioso debido a lo hostil del medio de comunicaciones. Por lo tanto, la integridad de los datos asegura de que los datos recibidos no hayan sido alterados en el trayecto.

A.3.3 Actualidad de los datos

Aún cuando se haya asegurado la confidencialidad e integridad de los datos, también se requiere asegurar la actualidad de cada mensaje. La actualidad de los datos sugiere que los datos sean recientes, y se asegura que ningún mensaje antiguo haya sido reenviado. Este requerimiento es especialmente importante cuando se utilizan estrategias de llaves compartidas en el diseño. Típicamente las llaves compartidas ocupan ser cambiadas a través del tiempo. Sin embargo, toma tiempo el propagar las nuevas llaves por toda la red. Bajo este esquema, seria fácil para un adversario el utilizar un ataque de reenvío de paquetes. También, seria fácil el corromper la operación normal de los nodos si es

que estos no están informados del tiempo en el que se cambiará la nueva llave. Para resolver este problema se puede agregar un contador dependiente del tiempo al paquete para asegurar la actualidad de los datos.

A.3.4 Autentificación

Un adversario no solo está limitado a modificar los paquetes. Potencialmente también podría modificar el flujo de los mismos al agregar paquetes adicionales al tráfico de la red. De tal manera que el receptor crea que los datos utilizados en cualquier proceso de toma de decisión de la red en realidad provienen de la fuente apropiada. Por otra parte, la autentificación es necesaria para varias tareas administrativas (por ejemplo: reprogramación de la red o controlar el ciclo de trabajo de los sensores). De lo dicho anteriormente, se puede observar que la autentificación de mensajes es importante para muchas aplicaciones en redes de sensores. Concretamente, la autentificación de datos permite al receptor verificar que los datos en realidad fueron enviados por el transmisor que dicen ser enviados. En el caso de la comunicación de dos nodos, la autentificación de datos se puede lograr mediante mecanismos simétricos: el transmisor y receptor comparten una llave secreta para calcular un código de autentificación del mensaje (MAC) de todos los datos por comunicar.

A.3.5 Disponibilidad

El hecho de ajustar los algoritmos de encriptación tradicional a redes inalámbricas de sensores implica un costo adicional. Algunos enfoques sugieren modificar el código de manera que se reutilice lo más posible. Otros enfoques tratan de utilizar comunicación adicional para lograr el mismo objetivo. Y otros enfoques más radicales aún, implantan

restricciones al acceso a datos o proponen esquemas poco robustos (como esquemas centralizados) para simplificar los algoritmos. Pero todos estos enfoques disminuyen el nivel de disponibilidad de los sensores y por lo consiguiente de la red en su totalidad por las siguientes razones:

- El introducir procesamiento adicional introduce también un consumo de energía adicional. Si se agota la energía en un sensor sus datos ya no estarían disponibles.

- También el introducir comunicación adicional consume más energía. Además el introducir mayor comunicación aumenta considerablemente la probabilidad de que se produzca un conflicto (por ejemplo: una colisión).

- Si se introduce un esquema centralizado se cuenta con un solo punto de falla. Esto es una amenaza latente para la disponibilidad de toda la red.

El requerimiento de seguridad no solo interfiere con las operaciones de la red, sino que también puede afectar de manera considerable la disponibilidad de toda la red.

A.3.6 Auto-configuración

Las redes inalámbricas de sensores son un caso extremo de redes ad hoc, las cuales requieren de que cada nodo sea independiente y flexible para auto-configurarse de acuerdo a diversas situaciones. No existe una infraestructura fija con el fin de administrar una red de sensores. Esto también trae consigo un gran reto para la seguridad de este tipo de redes. Por ejemplo, la dinámica de la red sugiere la idea de pre-instalar una llave compartida entre la estación base y todos los sensores (Eschenauer y Gligor, 2002). Varios esquemas de predistribución aleatoria de llaves han sido propuestos dentro del contexto de las técnicas de encriptación simétrica (Chan *et al.*, 2003; Eschenauer y

Gligor, 2002; Hwang y Kim, 2004; Liu *et al.*, 2005). Dentro del área de aplicación de la encriptación de llave pública en redes de sensores, esta misma dinámica exige mecanismos eficientes para la distribución de llaves. Así como las redes de sensores se deben de auto-configurar para llevar a cabo ruteo multi-saltos también se deben de auto-configurar para la administración de llaves y así establecer relaciones de confianza entre los nodos.

Si una red de sensores carece de auto-configuración, el daño producido por un atacante o inclusive por un ambiente hostil puede ser fatal.

A.4 Ataques a redes inalámbricas de sensores

La naturaleza de las redes de sensores las hace vulnerables a diversos tipos de ataques. Los ataques pueden ser lanzados en una variedad de formas, los más notables son los de negación de servicios (DoS), pero también existen los ataques de análisis de tráfico, violación de privacidad, ataques físicos y otros más. Los ataques de negación de servicios en redes inalámbricas de sensores van desde simplemente saturar el canal de comunicaciones de los nodos hasta ataques más sofisticados diseñados para violar el protocolo MAC 802.11 (Perrig *et al.*, 2004) o cualquier otra capa de la red de sensores.

Debido a las grandes diferencias de limitantes de energía y poder de procesamiento, el protegerse contra cualquier ataque de negación de servicios bien diseñado puede ser básicamente imposible. Un nodo más poderoso puede fácilmente bloquear a un nodo sensor normal y prevenir que este cumpla su función.

Observese que los ataques a redes inalámbricas de sensores no están limitados solamente a ataques de negación de servicios sino que también abarcan una gran variedad de ataques como nodos comprometidos, ataques a protocolos de ruteo y ataques físicos.

En la siguiente sección se presentarán los escenarios de ataque que potencialmente pueden enfrentar las redes inalámbricas de sensores seguido por una recopilación acerca de los ataques que ya han sido documentados en la literatura.

A.4.1 Escenarios de Ataque

Para proponer y desarrollar medidas de prevención y recuperación de ataques hacia redes de sensores es necesario conocer las características de los posibles atacantes. Los atacantes potenciales se pueden clasificar en dos grandes grupos(Karlof y Wagner, 2003): atacantes a nivel de motes y atacantes a nivel de computadora. En el primer caso, el atacante tiene acceso a nodos de sensores. Por otra parte, el atacante a nivel de computadora tiene acceso a dispositivos más poderosos tales como computadoras personales, PDAs, etc. Por lo tanto, en este caso, tienen grandes ventajas sobre los nodos legítimos: estos pueden tener una mayor fuente de energía, contar con procesadores más potentes, podrían tener también transmisores de alta potencia o una antena altamente sensible para escuchar tráfico.

Un atacante a nivel de computadora puede hacer más daño que un atacante que solo posee nodos de sensores. Por ejemplo, un nodo de sensor únicamente podría bloquear los enlaces de radio en su vecindad mientras que un atacante con una computadora portátil podría bloquear toda la red de sensores utilizando un transmisor más potente. Por otra parte un atacante a nivel de computadora podría potencialmente escuchar el tráfico de toda la red, mientras que el atacante a nivel de motes únicamente podría escuchar el tráfico en una área muy limitada.

Otra clasificación interesante de los atacantes es la que divide en atacantes externos y atacantes internos. En el texto anterior se discutieron ataques externos, que son donde

el atacante no tiene ningún tipo de acceso hacia la red de sensores. Por otra parte, los ataques internos son aquellos donde un participante autorizado de la red de sensores se ha tornado malicioso. Los ataques maliciosos pueden ser montados desde nodos comprometidos que se encuentran ejecutando código malicioso o desde computadoras portátiles que han tenido acceso a llaves de seguridad, código y datos de nodos legítimos.

A.4.2 Ataques a protocolos de ruteo

La gran mayoría de los protocolos de ruteo en redes de sensores son muy simples, debido a esto, en algunos casos son más susceptibles a ataques que los protocolos de ruteo para redes ad-hoc. La mayoría de los ataques de capa de red contra las redes de sensores caen en una de las siguientes categorías:

1. Información de ruteo falsificada, alterada o retransmitida.
2. Retransmisión selectiva.
3. Ataques de sumidero.
4. Ataques tipo Sybil.
5. Ataques tipo wormhole.
6. Ataques de desbordamiento de HELLOS.
7. Falsificación de señal de reconocimiento (ACK).

A continuación se aboradarán a un nivel más de detalle los ataques mencionados.

Información de ruteo falsificada, alterada o retransmitida

El ataque más directo en contra de los protocolos de ruteo es dirigido hacia la información de ruteo que es intercambiada por los nodos. Al falsificar, alterar o retransmitir la información de ruteo, los atacantes pueden crear ciclos de ruteo, atraer o repeler el tráfico de la red, extender o acortar rutas, generar mensajes falsos de error, particionar la red, aumentar la latencia de extremo a extremo, etc.

Retransmisión selectiva

Las redes multi-saltos funcionan suponiendo que los nodos retransmiten los mensajes recibidos de manera confiable. En un ataque de retransmisión selectiva, los nodos maliciosos se pueden negar a retransmitir ciertos mensajes y simplemente descartarlos, asegurándose así que estos no se propaguen en la red. Una modalidad simple de este ataque es cuando el nodo malicioso se comporta como un hoyo negro y se niega a retransmitir cada paquete que recibí. Sin embargo, esta modalidad de ataque es muy fácil de ser detectada ya que los nodos vecinos pueden concluir fácilmente que esa ruta no es válida y utilizar otra ruta. Una modalidad más sutil de lanzar este ataque es cuando el atacante selectivamente retransmite paquetes. De tal manera, que si un adversario está interesado en suprimir o modificar paquetes que vienen de cierta fuente lo puede hacer retransmitiendo el resto del tráfico y así no se levantaría sospecha alguna de tal ataque.

Ataque de sumidero

En un ataque de sumidero, la meta del adversario es la de atraer todo el tráfico de una sección en particular de la red por medio de un nodo comprometido, creando así un

sumidero metafóricamente hablando en el cual el adversario se encuentra justamente en el centro. Debido a que los nodos que se encuentran en la ruta en la que fluyen los paquetes tienen oportunidad de alterar los datos de aplicaciones, los ataques de sumidero pueden facilitar otro tipo de ataques (como ataques de retransmisión selectiva por ejemplo).

Los ataques de sumidero típicamente se presentan por medio de un nodo comprometido que resulta altamente atractivo para enrutar paquetes hacia el resto de la red. Esto se puede lograr falseando o retransmitiendo un anuncio de un enlace de alta calidad hacia una estación base. Algunos protocolos quizás traten de verificar la calidad de la ruta mediante paquetes de reconocimiento (ACK) de extremo a extremo. En este caso, un atacante a nivel de computadora puede proveer una ruta de alta calidad transmitiendo con la suficiente potencia para llegar de un solo salto a la estación base. Debido a la alta calidad de la ruta (ficticia o real como en el último caso) que contiene al nodo comprometido, es altamente probable que cada nodo vecino del nodo comprometido envíe paquetes a la estación base por medio de él, así como también, estos nodos propaguen lo atractivo de esa ruta falsa hacia otros nodos y así sucesivamente. De tal manera, que el nodo comprometido genera una amplia área de control, atrayendo todo el tráfico destinado a la estación base desde nodos a varios saltos de ella.

La principal motivación para lanzar un ataque de sumidero es que estos hacen que el ataque de retransmisión selectiva sea trivial ya que todo el tráfico de cierta área llega al nodo comprometido y de ahí, fácilmente el atacante puede suprimir o modificar paquetes que se originen en cualquier nodo de esa área.

Ataque tipo Sybil

En un ataque tipo Sybil (Douceur, 2002), un nodo presenta identidades múltiples a los demás nodos de la red. Cualquier red cuya funcionalidad esté basada en el hecho de que los nodos se comportarán correctamente se encuentra bajo el riesgo de un ataque tipo Sybil.

Los ataques tipo Sybil son una amenaza para los protocolos de enrutamiento basados en información geográfica. El ruteo consciente de la ubicación usualmente requiere el intercambio de información de las coordenadas para el ruteo eficiente de paquetes. Idealmente, se espera que cada nodo envíe solo un conjunto de coordenadas, pero con un ataque tipo Sybil un adversario pudiera estar en varios lugares simultáneamente.

Ataque tipo wormhole

En el ataque tipo wormhole(Hu *et al.*, 2002), el adversario forma un túnel virtual por medio de un enlace de baja latencia que toma los mensajes en una parte de la red y los retransmite en otra. El caso más simple de este ataque es cuando un nodo se encuentra en medio de otros dos nodos los cuales están retransmitiendo paquetes entre ellos. Sin embargo, los ataques tipo wormhole comúnmente involucran a dos nodos distantes coludidos para subestimar la distancia que hay entre ellos al relevar los paquetes mediante un canal de comunicación externo que únicamente esta disponible para el atacante.

En algunos casos, un adversario que se encuentre situado cerca de la estación base puede distorsionar completamente el enrutamiento mediante un ataque de este tipo. El adversario puede convencer a los nodos que típicamente utilizan múltiples saltos para llegar a la estación base de que ellos se encuentran a solo uno o dos saltos si utilizan

el nodo comprometido como parte de su ruta. Esto a su vez puede crear un sumidero como se mencionó anteriormente.

Los ataques de tipo wormhole típicamente se usan en combinación con retransmisión selectiva o con el de escuhar tráfico. La detección de este tipo de ataques es difícil cuando se lanza un ataque en conjunto con uno de tipo Sybil.

Ataque de desbordamiento de HELLOS

Algunos protocolos requieren que los nodos envíen paquetes HELLO de tipo broadcast para anunciarse a sus vecinos, y un nodo que recibe tal paquete supondría que se encuentra dentro de la zona de cobertura de radiofrecuencia del nodo que envió dicho paquete. Sin embargo, esta suposición podría ser falsa ya que un atacante a nivel de computadora podría enviar paquetes de este tipo con la suficiente potencia para convencer a todos los nodos de la red que el adversario es su vecino. Esto provocaría que los nodos cercanos traten de usar al adversario como ruta hacia la estación base mientras que los nodos lejanos tratarían de enviar mensajes directamente hacia el adversario, pero al ser su potencia de transmisión bastante menor a la del adversario, estos paquetes nunca llegarían a el, lo cual generaría un estado de confusión en la red de sensores.

Los ataques de desbordamiento de HELLOS pueden ser vistos como ataques tipo wormhole en un solo sentido ya que la ruta se anuncia por parte del adversario pero nunca es utilizada para retransmitir los mensajes.

Falsificación de señal de reconocimiento (ACK)

Algunos algoritmos de enrutamiento en redes de sensores se basan en el uso de señales de reconocimiento (ACK). En este caso, un atacante puede falsificar dicha señal en respuesta a los paquetes que este escucha. Esto lo que ocasiona es convencer al nodo

transmisor que un enlace débil es fuerte. Con esto, el adversario puede lanzar un ataque de retransmisión selectiva al falsificar las señales de reconocimiento hacia el nodo que se pretende atacar.

A.4.3 Ataques a Agregación de Datos

La agregación de datos en redes inalámbricas de sensores puede disminuir significativamente el sobrecosto de comunicación en comparación a que todos los nodos transmitan sus lecturas a la estación base. Sin embargo, la agregación de datos complica aún más el aspecto de la seguridad. Esto debido a que cada nodo intermedio puede modificar, falsificar o descartar mensajes o simplemente transmitir valores de agregación falsos, de tal manera que un solo nodo comprometido puede ser capaz de alterar de manera significativa el valor final agregado. Además de esto, la agregación de datos interfiere con la encriptación, ya que no es factible encriptar los mensajes utilizando una llave entre cada nodo y la estación base ya que los nodos intermedios deben de ser capaces de interpretar el mensaje para poder realizar la agregación.

A continuación se abordarán los problemas de seguridad que se han detectado en redes inalámbricas que utilizan mecanismos agregación de datos.

Ataques de un nodo intruso

Un atacante puede colocar nodos en la red de manera arbitraria, una vez logrado esto, los nodos intrusos pueden fácilmente escuchar tráfico transmitido y posteriormente modificarlo y reenviarlo. Los mensajes que pueden ser falsificados se ubican en dos grupos: mensajes de lecturas de nodos hacia los agregadores y resultado de agregaciones hechas por nodos intermedios. En ambos casos, la consecuencia es que se genere un valor de agregación final en la estación base que se encuentre muy alejado del valor real.

Sin embargo, la solución para este tipo de ataques es relativamente trivial y ya ha sido abordada en diversos trabajos (Hu y Evans, 2003). Se implementa un mecanismo de autentificación entre los nodos agregadores y los nodos finales (hojas del árbol de agregación) mediante la utilización de un Código de Autentificación de Mensaje (MAC) utilizando una llave secreta entre cada nodo y la estación base.

Ataques por nodos comprometidos

Un problema que presenta gran preocupación para los desarrolladores de aplicaciones seguras para redes inalámbricas de sensores es cuando un atacante obtuvo acceso a las llaves de seguridad existentes en los nodos, a estos nodos se les conoce típicamente como nodos comprometidos(Hartung *et al.*, 2005). Una vez logrado el acceso a las llaves secretas por parte del atacante, las medidas de seguridad mencionadas en la sección anterior se tornan insuficientes para satisfacer los requerimientos de seguridad.

El integrar protección física a los nodos de una red inalámbrica de sensores sea probablemente una solución infalible para abordar este tipo de amenazas, pero si se considera el escenario típico de una red de sensores, donde la red esta conformada por cientos o quizás miles de nodos, esto no resulta una solución viable desde el punto de vista económico.

Enseguida se mencionarán algunos de los ataques más específicos que se pueden generar a partir de comprometer nodos en una red inalámbrica de sensores que utiliza agregación de datos.

- **Negación de servicios**. Una vez que un atacante se haya apoderado de la estación base o un agregador, este puede lanzar un ataque de negación de servicios y negarse a responder a cualquier consulta (query).

- **Ataque tipo stealthy**. En un ataque tipo stealthy, la meta del atacante es hacer que el usuario acepte resultados de agregación falsos los cuales difieren significativamente de los resultados obtenidos de las lecturas reales sin que esto sea detectado por el usuario.

De tal manera, que el integrar mecanismos de seguridad que eviten o minimicen el daño causado a la red mediante este tipo de ataques representa un gran reto para la comunidad de investigadores en el área. Existen ya algunos trabajos preliminares que abordan este tema pero presentan serias limitaciones (Przydatek *et al.*, 2003). Primero que nada, los esquemas propuestos funcionan bajo ciertas suposiciones que no se satisfacen en la mayoría de los casos y segundo, estos trabajos no abordan el problema de la confidencialidad, la cual puede ser requerida por una gran gama de aplicaciones en un futuro no muy lejano.

A.4.4 Ataques físicos

Las redes de sensores operan típicamente en ambientes hostiles. En tales ambientes, el tamaño físico de los sensores aunado al modo de operación desatendido que ya se ha mencionado hacen a las redes de sensores altamente susceptibles a ataques físicos (por ejemplo: destrucción física de los nodos (Wang *et al.*, 2005b)). A diferencia de los ataques mencionados en secciones anteriores los ataques físicos destruyen a los nodos sensores permanentemente, por lo tanto las pérdidas son irreversibles. Por ejemplo, un atacante puede extraer llaves criptográficas, alterar la circuitería de un nodo, reprogramarlo, o reemplazarlo por nodos maliciosos (Wang *et al.*, 2005a). El trabajo presentado en (Hartung *et al.*, 2005) muestra que un mote MICA2 de Berkeley (los más utilizados actualmente por la comunidad científica) pueden ser comprometidos en menos de un

minuto. Aunque estos resultados no son sorprendentes ya que los motes MICA2 carecen de mecanismos de protección física, si nos dan una buena idea acerca de lo que puede hacer un atacante bien capacitado.

Apéndice B

Especificaciones técnicas de los motes MicaZ

Tabla III. Especificaciones técnicas de los motes MicaZ.

Especificación	**Valor**	**Comentarios**
Microcontrolador	ATmega128L	Corriendo a 4 MHz
Memoria Flash para programa	128K bytes	
Memoria Flash para datos	512K bytes	> 100,000 muestras
EEPROM para configuración	4K bytes	
Comunicación serie	UART	Niveles de voltaje de 0 a 3V
Convertidores analógico-digital	10 bits de resolución	8 canales, Entradas de 0 a 3V
Transreceptor RF	CC2420	Banda de los 2400 MHz a 2483.5 MHz
Tasa de transmisión	250 kbps	
Potencia de transmisión	-24 dBm a 0 dBm	
Alcance en exteriores	75 m a 100 m	
Alcance en interiores	20 m a 30 m	
Dimensiones (cm)	5.72 x 3.18 x 0.64	Excluyendo el compartimiento de baterías