



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistema de Monitoreo en Redes Híbridas del tipo
Infraestructura y Ad-Hoc con Sensores
Inalámbricos

Tesis que para obtener el título de

Ingeniero en Telecomunicaciones

Presenta

Fabiola Alavez Alvarez

Director de Tesis

Dr. Javier Gómez Castellanos



Ciudad Universitaria, México D.F.

Septiembre de 2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A Dios, esa fuerza inexplicable que me brindó el regalo de la vida, que me consuela con un atardecer, me fortalece con mi familia, y me ha dado la facultad suficiente para dar un paso más en mi educación realizando este trabajo.

A mi familia, a quienes quiero, respeto y admiro:

A mi padre, guía, consejero y amigo, que ha sido un ejemplo de sabiduría y voluntad ante las dificultades, y de afecto y generosidad para con la familia y amigos.

A mi madre, quien me ha dado el apoyo y cariño necesarios para afrontar cada reto, siendo para mí un modelo innegable de amor, paciencia y comprensión.

A mis hermanos, Carlos y Dan, quienes me han enseñado a sonreír y a buscar mi felicidad a pesar de los tropiezos.

A mis abuelitos, Hipólito e Ildelfonsa, Guadalupe y Amalia, por su sabiduría, consejos y voluntad, que han hecho de nuestras familias un lugar donde madurar, al amparo del amor, la confianza y la dignidad.

A Far Stein y Mor Astri, por mostrarme con su cariño que no necesitamos ser millonarios ni religiosos para convertirnos en filántropos.

A mis tíos(as), primos(as) y amigos(as) que han llenado mi vida de momentos de orgullo y felicidad, y sin cuya confianza y respaldo no podría lograr los retos que me he impuesto.

ÍNDICE GENERAL

<i>INTRODUCCIÓN</i>	1
<i>Capítulo 1</i>	2
<i>INTRODUCCIÓN A LAS REDES DE SENSORES</i>	3
1.1. ¿Qué son?.....	3
1.2. Nodo Sensor	3
1.3. Características básicas	4
1.4. Descripción de funcionamiento	5
1.5. Aplicaciones	6
<i>Capítulo 2</i>	8
<i>SISTEMA DE MONITOREO PARA OFICINAS</i>	8
2.1. Idea General.....	9
2.2. Necesidades en un entorno de oficina	9
2.3. Condiciones del entorno a monitorear.....	10
2.4. Requisitos del sistema de monitoreo.....	11
2.5. Modelo de red	11
2.5.1. Antecedentes.....	12
2.5.1.1. Redes con infraestructura	12
2.5.1.2. Redes sin infraestructura (Ad-Hoc).	13
2.5.1.3. Topología de una WSN	14
2.5.2. Estructura de la Red Implementada	15
<i>Capítulo 3</i>	18
<i>ENTORNO DE TRABAJO</i>	18
3.1. Redes de Sensores Cross-Bow	19
3.2. Documentación del Hardware y Software	21
3.2.1. PLATAFORMAS CROSS-BOW.....	22
3.2.2. SISTEMA OPERATIVO TinyOS.....	28
3.2.3. SOFTWARE DE APLICACIÓN	30
<i>Capítulo 4</i>	35
<i>VALIDACIÓN DEL SISTEMA</i>	35

<i>Capítulo 5</i>	47
<i>IMPLEMENTACIÓN DEL SISTEMA PROPUESTO</i>	47
5.1. Colocación de los sensores	49
5.2. Programación y Monitoreo.....	51
<i>Monitoreo con Mote-View</i>	52
5.3. Pruebas sobre el sistema	57
5.4. Aplicaciones del Sistema Implementado	63
<i>Capítulo 6</i>	65
<i>CONCLUSIONES</i>	65
6.1. Conclusiones.....	66
6.2. Contribuciones.....	66
6.3. Retos.....	68
6.4. Vías de Aplicación Futuras.....	69
<i>REFERENCIAS</i>	70
<i>APÉNDICE</i>	71
<i>Apéndice A: Funcionalidad Multi-Hop</i>	71
<i>Apéndice B: Componentes del Programa Surge_Reliable</i>	73

ÍNDICE DE FIGURAS

<i>Figura 1.1: Representación de una WSN.....</i>	<i>3</i>
<i>Figura 1.2: Nodo Sensor.....</i>	<i>4</i>
<i>Figura 1.3: Mote Mica2.....</i>	<i>4</i>
<i>Figura 1.4: Aplicaciones.....</i>	<i>6</i>
<i>Figura 2.1: Red con Infraestructura.....</i>	<i>12</i>
<i>Figura 2.2: Ejemplo de red Ad-Hoc.....</i>	<i>14</i>
<i>Figura 2.3: Topología en Árbol.....</i>	<i>14</i>
<i>Figura 2.4: Topología en Malla.....</i>	<i>15</i>
<i>Figura 2.5: Monitoreo de la Red desde una computadora con entrada serial.....</i>	<i>16</i>
<i>Figura 2.6: Monitoreo de la Red desde cualquier computadora con acceso a Internet.....</i>	<i>16</i>
<i>Figura 2.7: Monitoreo de la Red desde cualquier máquina en una red ethernet local.....</i>	<i>17</i>
<i>Figura 3.1: Bloques que constituyen una WSN.....</i>	<i>19</i>
<i>Figura 3.2: Series de Sensores, Motes y Puertas de Enlace de la compañía Cross-Bow.....</i>	<i>20</i>
<i>Figura 3.3: Kit de Sensores Cross-Bow (MOTE-KIT5040).....</i>	<i>21</i>
<i>Figura 3.4: MIB510CA.....</i>	<i>22</i>
<i>Figura 3.5: MIB510CA con Mica2DOT.....</i>	<i>23</i>
<i>Figura 3.6: MIB510CA con Mica2.....</i>	<i>23</i>
<i>Figura 3.7: Mote Mica2.....</i>	<i>23</i>
<i>Figura 3.8: Mote Mica2Dot.....</i>	<i>26</i>
<i>Figura 3.9: MTS310CA.....</i>	<i>26</i>
<i>Figura 3.10: MTS510CA.....</i>	<i>27</i>
<i>Figura 3.11: Surge-View : Topología de la Red.....</i>	<i>31</i>
<i>Figura 3.12: Surge-View : Tabla de estadísticas.....</i>	<i>32</i>
<i>Figura 3.13: SerialForwarder: Muestra la sincronización de los puertos.....</i>	<i>32</i>
<i>Figura 3.14: Ventana de interfase Mote-View.....</i>	<i>34</i>

<i>Figura 4.1: Estación de Monitoreo.</i>	<i>37</i>
<i>Figura 4.2: Gestor de Alertas de Mote-View.</i>	<i>38</i>
<i>Figura 4.3: Recuadro de Alertas.</i>	<i>39</i>
<i>Figura 4.4: Tabla de Estadísticas.</i>	<i>39</i>
<i>Figura 4.5: Topología de la Red para pruebas de envío de mensajes.</i>	<i>40</i>
<i>Figura 4.5: Patrón de Radiación de una Antena Monopolo.</i>	<i>42</i>
<i>Figura 4.6: Efectos de Polarización.</i>	<i>43</i>
<i>Figura 4.7: Caja protectora para los nodos sensores.</i>	<i>46</i>
<i>Figura 5.1: Componente Alámbrica de la Red.</i>	<i>48</i>
<i>Figura 5.2: Nodo Sensor.</i>	<i>49</i>
<i>Figura 5.3: Disposición de los Nodos Sensores dentro del Laboratorio de Pruebas.</i>	<i>50</i>
<i>Figura 5.4: Alarmas Establecidas.</i>	<i>56</i>
<i>Figura 5.5: Conectividad entre Nodos, Escenario 1.</i>	<i>58</i>
<i>Figura 5.6: Conectividad entre Nodos, Escenario 2.</i>	<i>60</i>
<i>Figura 5.7: Gráficas de Temperatura para los Nodos 0, 2 y 3.</i>	<i>61</i>
<i>Figura 5.8: Gráficas de Luz para los Nodos 5 y 6.</i>	<i>62</i>
<i>Figura 5.9: Topología de la Red; Niveles de Temperatura.</i>	<i>62</i>
<i>Figura 5.10: Topología de la Red; Niveles de Luz.</i>	<i>63</i>
<i>Figura A.1: Representación gráfica de la aplicación de TinyOS Surge_Reliable y Surge_Realiabile_Dot.</i>	<i>71</i>

ÍNDICE DE TABLAS

<i>Tabla 3.1: Relación entre el Consumo de Corriente y la Potencia de Salida en un radio CHIPCON CC1000.</i>	<i>25</i>
<i>Tabla 4.3: Alcance y Calidad de transmisión.</i>	<i>41</i>
<i>Q: Calidad del Enlace ; Y: Rendimiento de la Red.</i>	<i>41</i>
<i>Tabla 4.4: Calidad de la Señal.</i>	<i>44</i>

ÍNDICE DE GRÁFICAS

<i>Gráfica 4.1: Tiempo vs Luz.</i>	<i>36</i>
<i>Gráfica 4.2: Tiempo vs Temperatura </i>	<i>36</i>
<i>Gráficas 5.1: Calidad de Enlaces en la Red; Escenario 1.</i>	<i>57</i>
<i>Gráficas 5.2: Calidad de Enlaces en la Red; Escenario 2.</i>	<i>59</i>

INTRODUCCIÓN

La amplia disponibilidad de microprocesadores y dispositivos electrónicos cada vez más compactos y de menor costo comercial ha abierto hoy en día un número ilimitado de posibilidades en diversos campos de la investigación; uno de estos dominios son las Redes de Sensores Inalámbricos o WSN (Wireless Sensor Networking).

El campo emergente de las WSN combina la capacidad de sensar, procesar y comunicar datos todo en un pequeño dispositivo. A través de técnicas de enrutamiento y protocolos de comunicación estos pequeños dispositivos configuran por sí mismos una red completa y son capaces de trasladar la información a cualquier punto a través de los diferentes nodos que la componen.

La incorporación de sensores inalámbricos puede revolucionar la forma en que apreciamos y controlamos diversos sistemas físicos, desde el motor de una máquina hasta complejos sistemas electromecánicos, hidráulicos e incluso ecosistemas.

La capacidad de monitorear y manipular elementos físicos presenta infinidad de posibilidades en casi todos los campos de la ciencia; de manera que los escenarios de adaptación de estos sensores varían desde aplicaciones industriales, estudios ambientales o rastreo en tiempo real, hasta aplicaciones militares. Una fábrica de químicos puede utilizar estos dispositivos para monitorear tareas que podrían ser dañinas para un ser humano, de esta manera los dispositivos coleccionarían y enviarían los datos hasta un lugar seguro para los analistas. Dichos sensores también pueden ser esparcidos en selvas tropicales por ecologistas con el propósito de medir diariamente la temperatura, patrones de crecimiento, e inclusive patrones de migración de algunas especies de aves. Los datos recolectados y correlacionados por la WSN serán enviados por ésta a través de largas distancias hasta la comodidad y seguridad de sus laboratorios.

El propósito de este trabajo de investigación es plantear y desarrollar un sistema de monitoreo de oficinas utilizando una red de sensores inalámbricos. Se pretende estudiar la funcionalidad de los elementos que constituyen la red y llevar a cabo una validación inicial sobre la eficiencia de la red implementada. Para el desarrollo de este proyecto dispusimos del kit profesional MOTE-KIT5040 de la compañía Cross-Bow adquirido por el departamento de Telecomunicaciones de la Facultad de Ingeniería.

Este documento consta de seis capítulos; el primero de ellos es una breve introducción a las redes de sensores inalámbricas, se explican sus características básicas, funcionamiento y campos de aplicación. El segundo capítulo ofrece una descripción del sistema a implementar exponiendo las necesidades del entorno, los requisitos a cumplir y la metodología de resolución propuesta, así como los diferentes tipos de estructuras y topologías de red a utilizar. En el tercer capítulo se especifica el entorno de trabajo, es decir, las herramientas de hardware y software utilizadas en este proyecto. El cuarto capítulo explica la validación realizada sobre los componentes del sistema a implementar. En el quinto capítulo se detalla la implementación del sistema propuesto, es decir, la instalación, programación y monitoreo del sistema, así como las pruebas finales realizadas sobre el mismo. Por último en el sexto capítulo, en función de los resultados obtenidos, se extraen conclusiones sobre los objetivos alcanzados en esta investigación y se proponen vías futuras de desarrollo para proyectos posteriores.

Capítulo 1

INTRODUCCIÓN A LAS REDES DE SENSORES

1.1. ¿Qué son?

Las redes de sensores están compuestas por nodos que utilizan protocolos inalámbricos como medio de comunicación. La Figura 1.1 muestra una representación esquemática de una WSN. Cada nodo tiene múltiples enlaces inalámbricos con otros nodos y eventualmente se conectan a una estación base. Dicha estación sirve como puerta de enlace hacia otras redes, como por ejemplo Internet, o hacia un servidor desde el cual se lleva a cabo el monitoreo de la red de sensores.

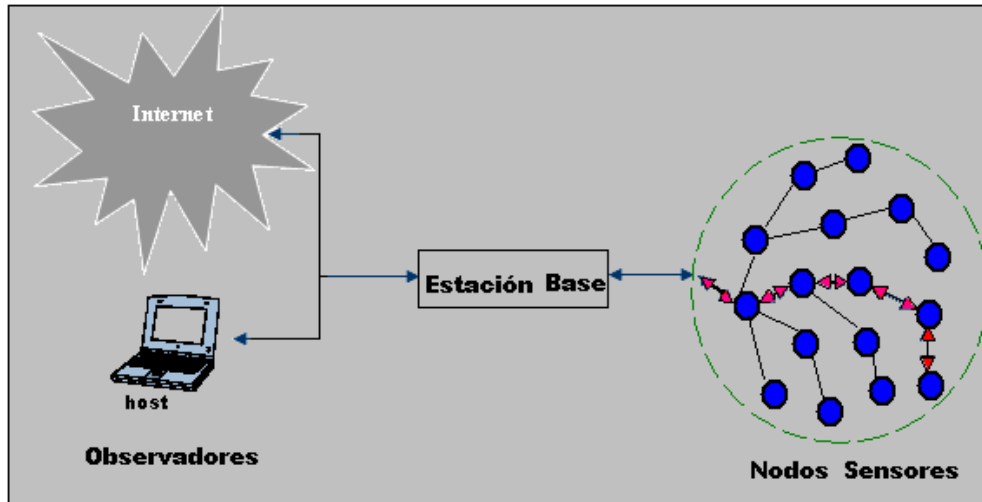


Figura 1.1: Representación de una WSN.

Cada nodo tiene uno o varios sensores que pueden detectar cambios en el entorno, como son: temperatura, presión, humedad, luz, velocidad, sonido, etcétera. El reporte y procesamiento de los datos se mueve de un nodo a otro desde el punto en que se origina el evento, hasta el punto en donde se utiliza dicha información.

1.2. Nodo Sensor

Un nodo se compone de cuatro unidades principales:

- 1) **Unidad de sensado.** Incluye los sensores (que serán los ojos y oídos del nodo) y un convertidor A/D que transforma la señal analógica en un flujo de datos digital, para después conducirlo a la unidad de procesamiento.
- 2) **Unidad de procesamiento.** Se compone de un microprocesador y una pequeña unidad de memoria; su función es el manejo de los procesos que permiten la colaboración entre los diferentes nodos.

- 3) **Unidad de transmisión-recepción.** Es generalmente un radio de RF¹ que conecta al nodo con sus nodos vecinos.
- 4) **Unidad de potencia.** No es más que una pequeña batería.

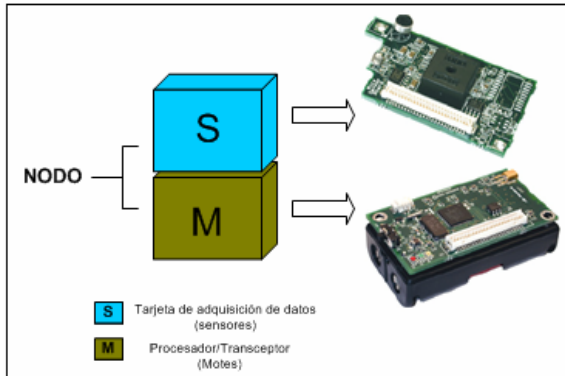


Figura 1.2: Nodo Sensor.



Figura 1.3: Mote Mica2.

La Figura 1.2 muestra uno de los nodos utilizado en este proyecto. Al mismo tiempo, en la Figura 1.3, podemos observar el procesador y la unidad de transmisión-recepción del nodo; ambos integran la plataforma llamada mote Mica2, la cual es uno de los productos comerciales de Cross-Bow y será descrita ha detalle en capítulos posteriores.

Debido a que cada nodo es inalámbrico y generalmente opera por largos periodos de tiempo, se tiene un recurso de energía restringido; esto obliga a que cada nodo tenga un consumo de potencia muy bajo.

Con el fin de garantizar el bajo consumo de energía, los nodos tienen restricciones en cuanto a velocidad de procesamiento, capacidad de almacenamiento y ancho de banda de transmisión. Para compensar estas limitantes una WSN debe tener una densidad de nodos muy alta y contar con un alto grado de interacción entre ellos.

1.3. Características básicas

Esta clase de red se caracteriza por su facilidad de despliegue y por ser autoconfigurable. Cada uno de los dispositivos sensores puede convertirse en todo momento en emisor o receptor, ofrecer servicios de enrutamiento entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Otra de sus características es la gestión eficiente de energía, lo cual ofrece a los nodos un alto grado de autonomía. Algunas de las funciones que permiten estas facilidades son:

¹ Las frecuencias de radio utilizadas se encuentran dentro de las bandas ISM (Industrial, Scientific and Medical)

- **Comunicación Multisalto.** Facultad de enviar mensajes peer-to-peer hasta una estación base, permitiendo de ese modo la escalabilidad en grandes rangos de extensión.
- **Auto configuración.** Habilidad para formar redes de trabajo sin intervención humana.
- **Auto reparación.** Permite adherir o remover nodos de la red de manera automática sin necesidad de resetearla; y
- **Ruteo Dinámico.** Capacidad de adaptar la red y determinar la ruta de información, basándose en las condiciones dinámicas de la misma (calidad de enlace, contador de saltos, etcétera.)

Como se mencionó anteriormente; el combinar estas características con un uso adecuado de la energía existente (un par de baterías) permite a la red tener un periodo de vida mayor, ser de fácil implementación y resistente a las variantes en el canal de transmisión inalámbrico. Estas características hacen realidad la visión de una red de sensado con un número de nodos casi ilimitado.

1.4. Descripción de funcionamiento

La red de trabajo de una WSN funciona a través de las primeras cinco capas del modelo de referencia OSI: Capa física, Capa de enlace de datos, Capa de red, Capa de transporte y Capa de aplicación.

La capa física controla el radio del nodo, es decir, la selección de la frecuencia de operación, generación de la portadora de frecuencia, detección de la señal y encriptación de los datos enviados.

La capa de enlace de datos se ocupa del direccionamiento físico, de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control del flujo de datos binario. Esto asegura una conexión punto-multipunto confiable en la red de comunicación. En otras palabras, es la encargada de escuchar el canal y transmitir cuando éste se encuentre desocupado.

El cometido de la capa de red es hacer que los datos lleguen desde el origen al destino; es decir, se encarga del enrutamiento o ruteo de los datos provenientes de la capa de transporte. Adicionalmente, la capa de red debe gestionar la congestión de la red.

La función básica de la capa de transporte es aceptar los datos enviados por las capas superiores, segmentarlos si es necesario, y pasarlos a la capa de red.

Finalmente, dependiendo de las tareas de sensado que se requieran, se pueden construir diferentes tipos de aplicaciones, el software requerido puede construirse y usarse sobre la capa de aplicación. Esta capa ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos.

1.5. Aplicaciones

Las redes de sensores son sistemas de monitoreo en línea que permiten conocer en tiempo real las condiciones de operación y el estado del sistema con el objetivo principal de detectar problemas incipientes, tomar decisiones oportunas y reducir la probabilidad de fallas catastróficas en equipos o sistemas. La diversidad de aplicaciones de las WSN es muy grande; básicamente podríamos agruparlas de la siguiente manera:

- **Monitoreo de un espacio.** Estudio de un ecosistema, medio ambiente o hábitat; agricultura de precisión, observación y control de micro ecosistemas, control del clima en interiores, vigilancia, etcétera.
- **Supervisión de entidades.** Mantenimiento de equipos y estructuras basándose en su condición, diagnósticos médicos, cartografía de terrenos urbanos, etcétera.
- **Registro de la interacción entre objetos o la interacción de éstos con su entorno.** El hábitat de vidas silvestres, control de desastres, respuesta a emergencias, sistemas de seguimiento, cuidado de la salud, control del flujo en procesos de manufactura, etcétera.
- **Aplicaciones Militares.** Resguardo de fronteras, sensores a distancia para seguir los movimientos enemigos, detección de gas venenoso ó radiactividad, etcétera.



Figura 1.4: Aplicaciones.

Algunas de las ventajas que provee una WSN sobre las tecnologías de sensado utilizadas actualmente son:

- La infraestructura requerida es mínima.
- Los componentes de la red requieren de poca energía para su funcionamiento.
- El hardware utilizado es de bajo costo.
- Es fácil de desarrollar, instalar y mantener.
- Reduce costos de cableado y tiempo de instalación.
- Permite movilidad, portabilidad y regeneración de la red.
- Transmisión de datos que permite un control de realimentación remoto.
- Genera mejoras en la productividad.

Capítulo 2

SISTEMA DE MONITOREO PARA OFICINAS

2.1. Idea General

El objetivo de este proyecto de investigación es plantear y desarrollar un sistema de monitoreo de oficinas utilizando una red de sensores inalámbricos; estudiar la funcionalidad de los elementos que constituyen la red y llevar a cabo una validación inicial del sistema propuesto.

Para delimitar las funciones de nuestro sistema se establecieron los siguientes puntos:

- Necesidades ante las que el sistema debía responder,
- Condiciones del entorno a monitorear,
- Requisitos del sistema de monitoreo, y
- Modelo de red a utilizar.

2.2. Necesidades en un entorno de oficina

Se plantearon tres necesidades básicas:

1. Control de temperatura

El sistema debería desplegar una alarma en los siguientes casos:

- a) Si la temperatura en la oficina es menor a 20.56 [° C] o mayor a 20.56 [° C].
- b) Si la temperatura en la oficina es mayor a 23.30 [° C] o menor a 23.30 [° C].

El primer umbral establece las temperaturas de encendido y apagado para un sistema de calefacción, mientras que el segundo las determina para un sistema de aire acondicionado. De manera que estos sistemas podrían anexarse fácilmente al sistema implementado en este proyecto.

Como podemos observar, los umbrales definidos establecen una zona de confort que oscila entre los 20.56 y 23.30 [° C] de temperatura. Estos umbrales se fijaron tomando en cuenta las recomendaciones sobre “zonas de confort”, citadas por Olgay en su libro *Arquitectura y Clima*¹, y las cuales están dentro de la norma internacional ISO7730².

2. Control de iluminación

El sistema debería desplegar una alarma si se detectaba un nivel de intensidad menor a 200 [lux]. Esta alarma representa el umbral a seguir por un sistema de iluminación; el cual puede adicionarse al sistema propuesto.

¹ OLGAY Víctor. *Arquitectura y Clima*. Manual de Diseño Bioclimático para Arquitectos y Urbanistas, tr. del inglés por Josefina Frontado y Luis Clavet. Barcelona: Gustavo Gili, 1998.

² Introducción a la conferencia internacional: *Moving Thermal Comfort Standard into the 21st Century*. (<http://www.brookes.ac.uk/schools/arch/res/ocsdcon.html#Intro>).

El nivel de intensidad se consideró tomando en cuenta una exigencia visual moderada en edificios de oficinas³.

3. Vigilancia

El sistema debería activar una alarma en caso de detectar sonidos o movimiento fuera del horario de labores en la oficina.

2.3. Condiciones del entorno a monitorear

- La oficina se trata de un lugar cerrado en el que no existe una instalación previa de control de clima.
- El horario de operación, tanto de la oficina como del sistema, se fijaron de ocho de la mañana a nueve de la noche.
- Las condiciones climáticas a las que se sujeta el entorno de monitoreo se encuentran dentro de los siguientes rangos⁴:
 - Temperatura máxima en verano de 28 [° C]
 - Temperatura mínima en invierno de 18 [° C]
- Se tomaron en cuenta la ubicación de paredes, muebles y equipo disperso en el entorno, los cuales obstaculizan el envío de señales entre los nodos sensores. Debido a estos elementos la señal de RF decae más rápidamente en lugares cerrados que en lugares abiertos, es decir, la cobertura se encuentra delimitada por barreras físicas. Estos, entre otros factores, fueron pautas que sirvieron para ubicar los sensores dentro del escenario de prueba.
- La oficina tiene las siguientes dimensiones: 7 metros de largo, 7 metros de ancho y 3.5 metros de altura. Cabe mencionar que en una extensión mayor las condiciones de clima, interferencias existentes, posicionamiento de muebles, etcétera, pueden variar considerablemente y con ellos la manera en que la red debe extenderse.
- Otro factor que afecta el comportamiento de la señal de RF en interiores es la existencia de otras señales de radio. El entorno en el que se trabajó no mostró interferencias considerables de este tipo, sin embargo cabe mencionarlo ya que existen lugares muy propensos a ellas. El medio en que se propaga nuestra señal podría estar atestado de señales dentro de la misma banda; por ejemplo una oficina de un hospital en donde varios dispositivos médicos trabajan en la banda ISM.

³ Hernández Chávez, Vicente, *La habitabilidad energética en edificios de oficinas*, 2002.
(http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0109103-155648/)

⁴ Ciudad de México: 2000 [m] sobre el nivel del mar.

2.4. Requisitos del sistema de monitoreo

La concepción y desarrollo de un sistema de monitoreo de oficinas usando una red de sensores inalámbricos supone un conjunto de requisitos técnicos y tecnológicos a satisfacer:

- a) Contar con una estación central encargada de recibir las señales. Aquí se llevará a cabo el procesamiento y visualización así como la emisión de las alarmas correspondientes.
- b) Avanzar rápidamente de la etapa de detección del problema a la de resolución.
- c) Adquisición, digitalización y transmisión de la señal en tiempo real, para ello debe minimizarse el retraso máximo de los mensajes.
- d) Transmisión de las señales de RF hasta la estación base; el alcance de cada nodo sensor no debe ser menor a 5 [m]. Esta distancia se fija en base a los requerimientos del espacio de prueba. En un espacio de 7x7 [m] una distancia de alcance menor a cinco metros supondría un mayor número de nodos y consecuentemente un mayor costo.
- e) Obtención de señales de alta calidad, para ello es necesario contar con una forma de transmisión altamente inmune y cumplir con estándares internacionales de transmisión para evitar ser interferido o interferir con otros dispositivos.
- f) Es deseable que la red pueda seguir funcionando (quizás de forma degradada) ante el fallo de alguno de sus nodos y/o enlaces. Por ello, la fiabilidad de la red es una de las características más importantes. Algunos de los parámetros propuestos para medir dicha fiabilidad son la conectividad y vulnerabilidad de la red
- g) Los módulos utilizados deben ser resistentes y de fácil mantenimiento, esto para garantizar un buen funcionamiento en condiciones de trabajo que pudieran ser adversas.
- h) Adicionalmente, debe ser fácil para un usuario con pocos conocimientos de computación o programación, extraer datos significativos del mundo real con sólo un entrenamiento básico de la red.

2.5. Modelo de red

En esta sección se describen los elementos teóricos que fundamentan el modelo de red utilizado en este proyecto. Estos elementos permiten comprender el funcionamiento de los componentes de la red implementada y a su vez percibir las ventajas sobre otro tipo de estructuras.

La topología establece la distribución de los dispositivos que conforman la red. Ésta depende del número de nodos a conectar y del tipo de acceso al medio que se desee. La flexibilidad de una red en cuanto a sus necesidades futuras se refiere, depende en gran parte de la topología establecida.

La estructura de la red determina la forma en que será transferida la información una vez que ésta ha viajado desde el lugar en que se produce el evento hasta una unidad recolectora, es decir, determina el método más conveniente para que la información recolectada llegue a manos de sus observadores o analistas.

2.5.1. Antecedentes

La complejidad de la configuración física de una red inalámbrica puede ser muy variada, dependiendo de las necesidades que se presenten y requerimientos del sistema a implementar. Básicamente se pueden clasificar en dos grandes grupos: Redes con infraestructura y Redes sin Infraestructura.

2.5.1.1. Redes con infraestructura

Una topología de infraestructura es aquella que amplía una red cableada existente, incorporando dispositivos inalámbricos mediante el uso de una estación base denominada punto de acceso. El punto de acceso une la red inalámbrica con la cableada y sirve como controlador central de la red inalámbrica. La extensión de la red inalámbrica y el número de dispositivos dependen del estándar de conexión inalámbrico que se utilice. En este tipo de topología podemos tener varios puntos de acceso para dar cobertura a una zona grande o un único punto de acceso para una zona pequeña como una casa o un edificio.

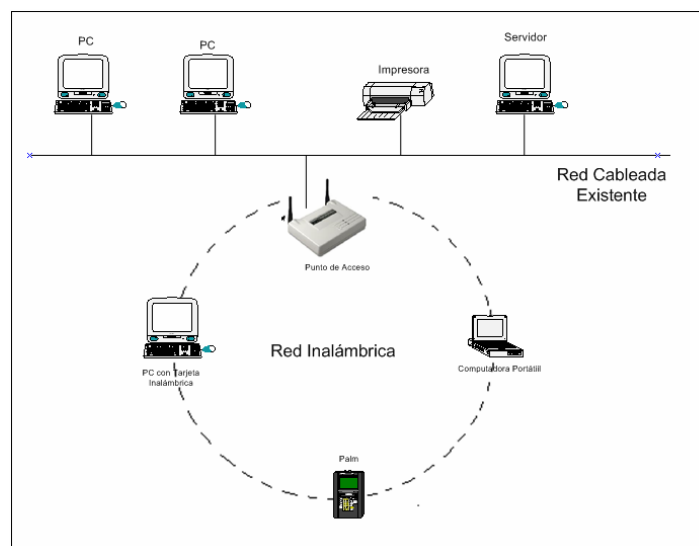


Figura 2.1: Red con Infraestructura.

Funcionamiento

El dispositivo inalámbrico primero debe identificar los puntos de acceso y las redes disponibles. Este proceso se lleva a cabo mediante el control de las tramas de señalización procedentes de los puntos de acceso que se anuncian a sí mismos, o mediante el sondeo activo de una red específica con tramas de sondeo.

La estación elige una red entre las que se encuentran disponibles e inicia un proceso de autenticación con el punto de acceso. Una vez que el punto de acceso y la estación se han verificado mutuamente comienza el proceso de asociación. Este proceso permite que el punto de acceso y el dispositivo inalámbrico intercambien información y datos de capacidad. El punto de acceso puede utilizar esta información y compartirla con otros para diseminar la información de la ubicación actual del dispositivo inalámbrico en la red. Este dispositivo sólo puede transmitir o recibir tramas después de que haya finalizado la asociación.

En la modalidad de infraestructura, todo el tráfico procedente de los dispositivos inalámbricos pasa por un punto de acceso para poder llegar a su destino, ya sea en la red cableada o en la red inalámbrica. De manera que tanto el emisor como el receptor son siempre el punto de acceso. La sincronización entre los dispositivos inalámbricos que conforman la red se controla mediante tramas de señalización periódicas enviadas por el punto de acceso. Estas tramas contienen el valor de reloj del punto de acceso en el momento de la transmisión, el cual sirve para comprobar la evolución en el dispositivo receptor.

2.5.1.2. Redes sin infraestructura (Ad-Hoc).

Una topología sin infraestructura está formada por dispositivos móviles que se comunican cada uno directamente con los otros a través de señales de radio. La red se forma de manera dinámica y arbitraria y no requiere de la ayuda de ninguna infraestructura externa. Es decir, no hay ningún elemento fijo o punto de acceso y la topología de la red puede adoptar múltiples formas conservando su funcionalidad.

En este tipo de redes todos los nodos funcionan como enrutadores móviles (*routers*) y se ven involucrados tanto en el descubrimiento como en el mantenimiento de rutas. Los dispositivos de la red inalámbrica que requieran comunicarse entre ellos necesitan usar el mismo canal de radio y configurar un identificador específico de WiFi en Modo *Ad Hoc*.

Funcionamiento.

En esta red sólo hay dispositivos inalámbricos presentes. Muchas de las operaciones que controlaba el punto de acceso, como la señalización y la sincronización, son ahora controladas por cada nodo.

La Figura 2.2 muestra una sencilla red *Ad-Hoc*. El nodo *origen* quiere enviar un paquete al nodo *destino*; debido a que éste se encuentra fuera de su alcance de transmisión (representado por círculos en la figura), se requieren de nodos intermedios que retransmitan el paquete. El responsable de descubrir las rutas entre los nodos para hacer posible la comunicación es el protocolo de enrutamiento.

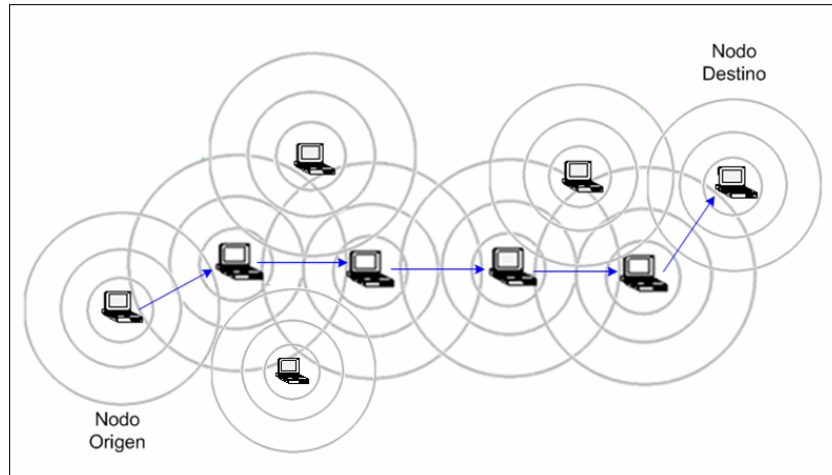


Figura 2.2: Ejemplo de red Ad-Hoc.

2.5.1.3. Topología de una WSN

La topología física de la red es la forma en que se distribuyen los nodos para comunicarse entre ellos y con el servidor. Las redes Ad-Hoc utilizan diferentes topologías físicas: de estrella, árbol, malla, etcétera.

En una topología de árbol se tiene un nodo de enlace troncal desde el que se ramifican los demás nodos, la información avanza de manera jerárquica a través de ellos hasta llegar al nodo recolector ubicado en la puerta de enlace (Figura 2.3).

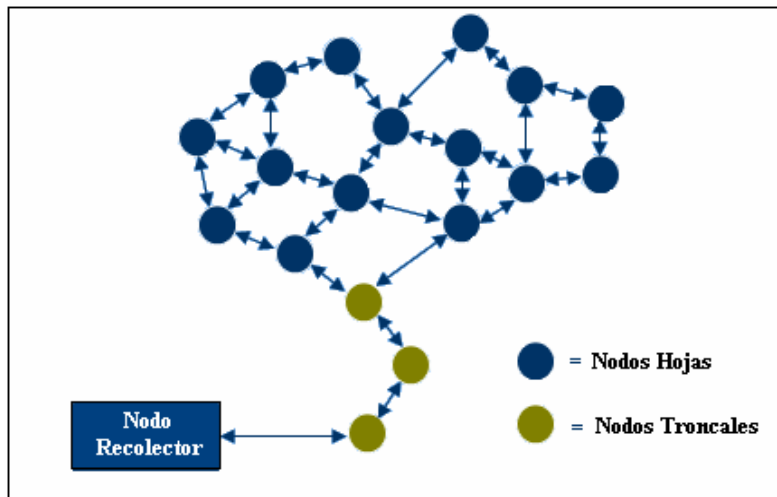


Figura 2.3: Topología en Árbol.

Una de las desventajas principales de este tipo de topología, es que los nodos troncales tienen un desgaste de energía mucho mayor al de los nodos ubicados en sus ramificaciones, esto se debe a que necesitan transmitir con mayor constancia; dando como resultado una reducción de su tiempo de vida.

En una topología de malla (Figura 2.4), cada nodo se enlaza directamente con al menos otros dos nodos, permitiendo así cierto nivel de redundancia en el sistema. Si algún nodo falla los datos pueden circular hasta el nodo recolector a través de otros nodos, de manera que sólo el nodo descompuesto es eliminado de la red.

Este tipo de topología también asegura una distribución uniforme del uso de energía entre todos los nodos, dado que la información tiene varias posibilidades de rutas para circular dentro de la red.

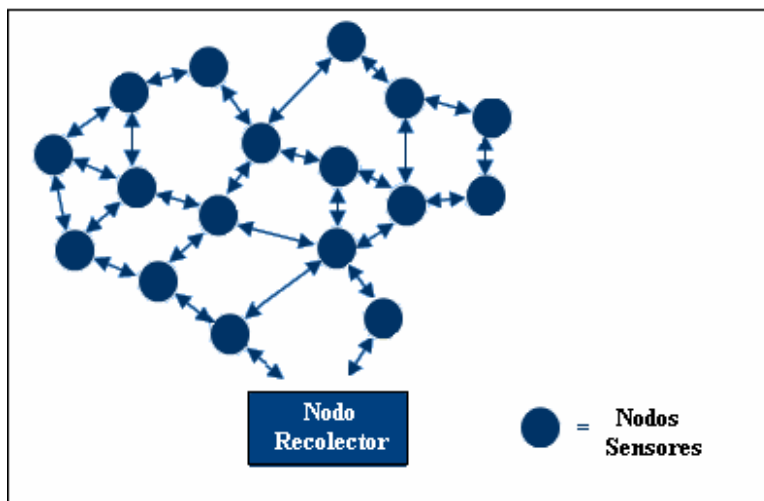


Figura 2.4: Topología en Malla.

La desventaja principal de la topología en malla es el costo que implica una alta densidad de nodos, ya que cada nodo debe estar conectado con al menos otros dos. Esto implica un costo de desarrollo elevado, sin embargo, si la prioridad del usuario es la redundancia y robustez del sistema, el incremento del costo puede parecer insignificante ante estos requerimientos principales.

2.5.2. Estructura de la Red Implementada

Las redes de sensores forman en esencia una red Ad-Hoc ya que los nodos sensores se comunican cada uno directamente con los otros a través de señales de radio, sin embargo, para poder monitorear nuestra red desde diversos puntos, es necesario contar con un gateway o puerta de enlace que conecte la red fija con la inalámbrica. Al incluir uno o

varios gateways estamos desplegando cierta infraestructura, por lo que decimos que se trata de una red Ad-Hoc Híbrida.

La puerta de enlace puede utilizar dos métodos diferentes de transferencia de datos; serial o ethernet. En el primer caso la conexión serial a una computadora nos permite monitorear la red desde esta Terminal, tal como se observa en la Figura 2.5.

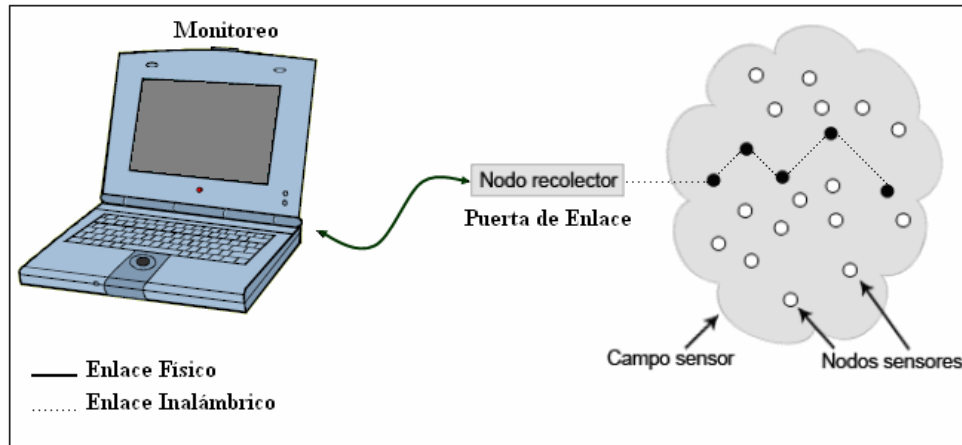


Figura 2.5: Monitoreo de la Red desde una computadora con entrada serial.

En el caso de que dicha computadora tenga acceso a Internet, Figura 2.6, la red puede ser monitoreada desde cualquier punto que cuente con este tipo de acceso.

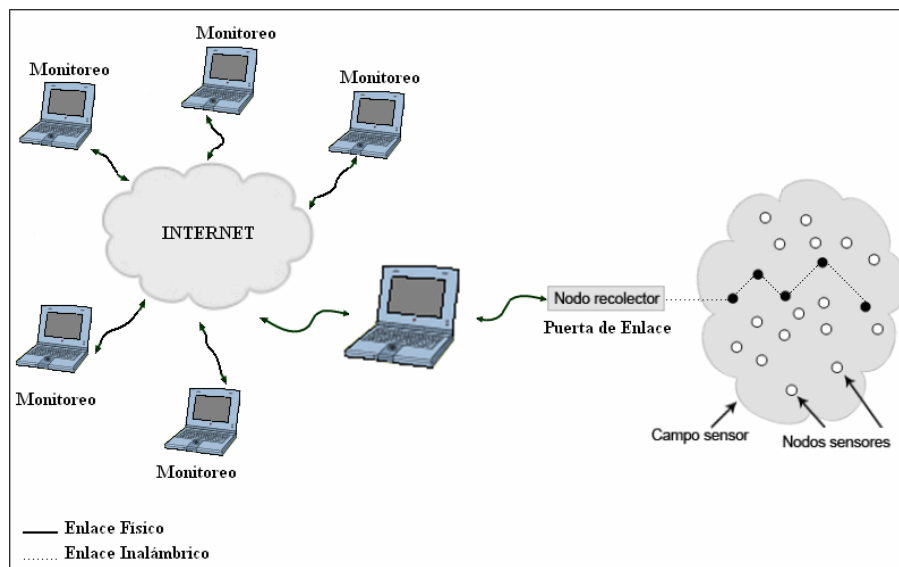


Figura 2.6: Monitoreo de la Red desde cualquier computadora con acceso a Internet.

Si los datos se transfieren a través de una conexión ethernet, la puerta de enlace debe tener una IP asignada, de manera que esta transfiera los datos a través de la red TCP/IP hasta otro elemento dentro de la red. La puerta de enlace puede estar conectada a un hub o switch por medio de un cable directo o a una computadora mediante un cable cruzado.

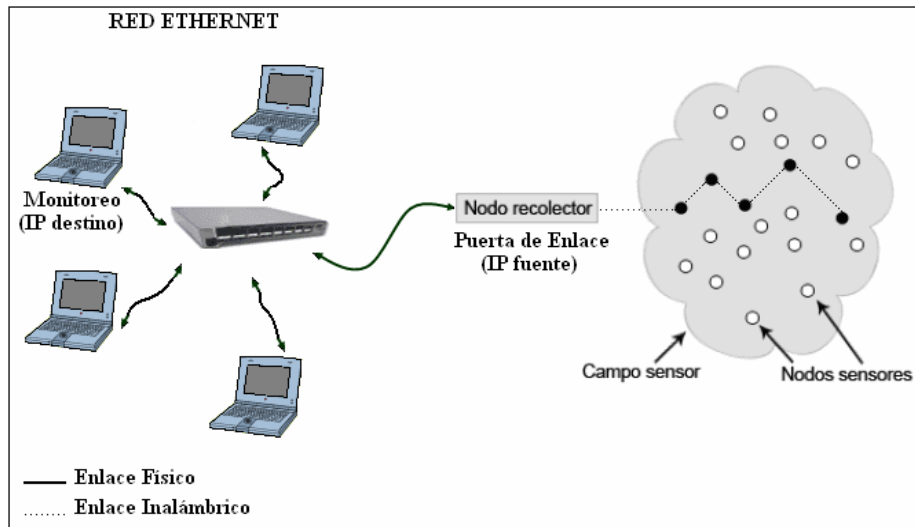


Figura 2.7: Monitoreo de la Red desde cualquier máquina en una red ethernet local.

En este proyecto se utilizó una red Ad-Hoc Híbrida del tipo mostrado en la Figura 2.5, y dentro del campo de sensores una topología en malla. El modelo elegido permite:

- Evitar la congestión en un punto único de la red.
- Reducir el número de saltos entre un nodo y otro.
- Redundancia. Intenta tener múltiples enlaces a la red a través de distintos nodos. Si uno falla se pueden usar los otros.
- Utilizar herramientas para monitorizar la red y así poder prevenir problemas futuros.
- Evitar las configuraciones manuales y utilizar configuraciones estándar para los diferentes componentes instalados.
- Mantener una buena comunicación entre los gestores de los nodos. La comunicación entre gestores es muy importante pues minimiza el tiempo necesario para resolver problemas que pudieran.

Capítulo 3

ENTORNO DE TRABAJO

En este capítulo se describen las herramientas de hardware y software utilizadas en la presente investigación. En el laboratorio trabajamos con un equipo Windows XP y un kit de desarrollo comercial de Cross-Bow. En cuanto al software, se utilizó el sistema operativo TinyOS, especialmente diseñado para las características intrínsecas de los nodos sensores y sus aplicaciones de red.

3.1. Redes de Sensores Cross-Bow

Como se mencionó en el Capítulo 1, una red de sensores se compone de un gran número de nodos esparcidos en el área a monitorear, dichos nodos tienen la capacidad de coleccionar y enviar los datos de nodo en nodo hasta una estación base que sirve como puerta de enlace de toda la red.

Actualmente existen varias WSNs disponibles en el mercado comercial. Una de las compañías líderes en este campo es Cross-Bow, la cual ofrece una diversa línea de productos para WSNs diseñados para diferentes escenarios. El bloque principal de construcción para WSNs que ofrece Cross-Bow puede dividirse en tres categorías: Sensores, Plataformas con Procesador y Radio (comúnmente conocidas como motes) y Plataforma de interfase, las cuales funcionan como puerta de enlace o estación base de la red. La Figura 3.1 muestra la forma en que se puede implementar una red utilizando los bloques mencionados.

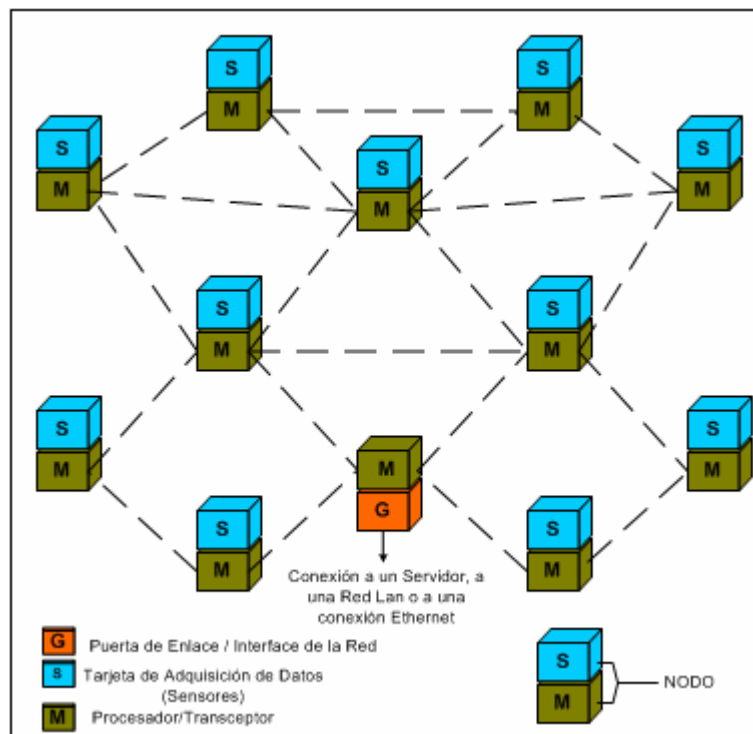


Figura 3.1: Bloques que constituyen una WSN.

Cada sensor y mote se acoplan para formar un nodo. Este nodo puede comunicar y transmitir datos a otros nodos o a la estación base. La estación base o puerta de enlace retransmite los datos recibidos ya sea a la computadora local, a una red LAN o a la red de Internet. Actualmente Cross-Bow ofrece la siguiente gama de sensores en el mercado comercial:

- **Dos series de sensores:** Series MicaX y Series MicaXDOT. Éstas a su vez se dividen en sub-series de acuerdo a su tamaño, frecuencia de operación, etcétera.
- **Tres series de motes:** Series Mica2, Series Mica2DOT y Series MicaZ. Los motes se categorizan de esta manera debido a factores de compatibilidad, ya que no todos los motes son compatibles con todas las tarjetas de sensores y viceversa.
- **Tres diferentes tipos de puertas de enlace:** MIB510, MIB600 y Stargate. Éstas difieren, entre otras cosas, en el método de transferencia de datos, frecuencias de operación, procesador utilizado, etcétera. Por ejemplo; la tarjeta MIB510 ofrece solamente una conexión RS-232 para la transferencia de datos, mientras que la tarjeta MIB600 ofrece conexión Ethernet (TCP/IP) y el tercer tipo, Stargate, ofrece la conexión RS-232 y también conexiones Ethernet e Inalámbrica ó Wireles (802.11 a/b).

La Figura 3.2 muestra una ilustración de las tarjetas de sensores, motes y puertas de enlace descritas.

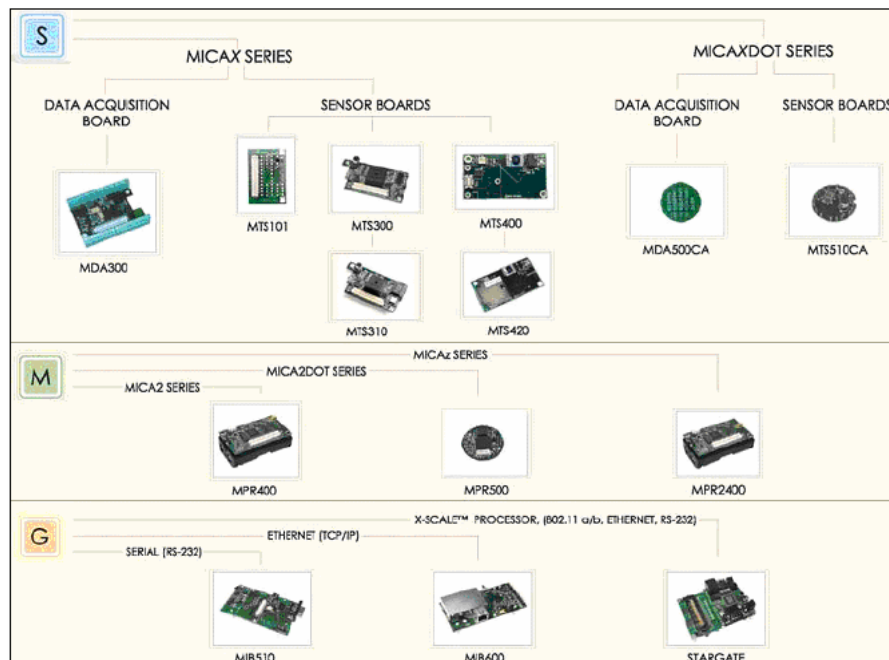


Figura 3.2: Series de Sensores, Motes y Puertas de Enlace de la compañía Cross-Bow.

Para la elaboración de este trabajo de investigación se utilizó una pequeña WSN de Cross-Bow: el kit profesional MOTE-KIT5040. Este kit es una combinación de los siguientes dispositivos:

- tres sensores MTS310CA (luz, temperatura, acústica y resonador),
- dos sensores MTS510 (aceleración, luz, acústica),
- cuatro motes MPR400 de la serie Mica2,
- cuatro motes MPR500 de la serie Mica2DOT, y
- una puerta de enlace MIB510.

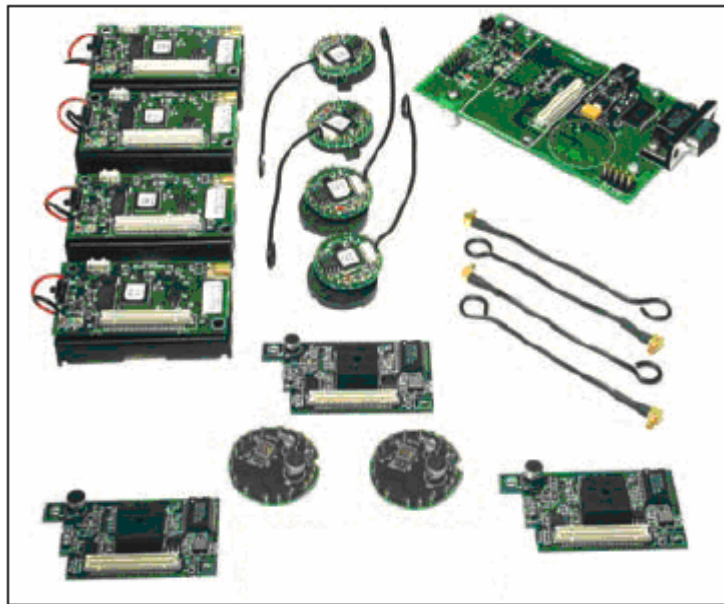


Figura 3.3: Kit de Sensores Cross-Bow (MOTE-KIT5040).

3.2. Documentación del Hardware y Software

En este apartado se describen las funciones y características técnicas de cada uno de los dispositivos que conforman el kit profesional MOTE-KIT5040. Se especifican además, el modo de empleo y las precauciones a considerar en este tipo de equipos.

En cuanto a software utilizado, se presenta una breve descripción del Sistema Operativo TinyOS, el cual es la plataforma de desarrollo que utilizan los dispositivos sensores para conformar diversas aplicaciones de red.

Finalmente, se mencionan las cualidades del software de aplicación Surge-View y Mote-View implementados por la compañía Cross-Bow. Éstos se encuentran diseñados sobre

la plataforma TinyOS y permiten de manera práctica y eficaz el monitoreo de una red de sensores.

3.2.1. PLATAFORMAS CROSS-BOW

➤ Puerta de enlace MIB510CA

La puerta de enlace MIB510CA permite retransmitir los datos recibidos a una computadora local mediante un puerto serie RS-232. Cualquier plataforma Mica2 puede funcionar como una estación base cuando es interconectada con la puerta de enlace MIB510CA.

Además de la transferencia de datos, la interfase serial RS-232 nos proporciona el enlace necesario para que las aplicaciones desarrolladas en la computadora puedan ser programadas en los motes Mica2 y Mica2DOT.

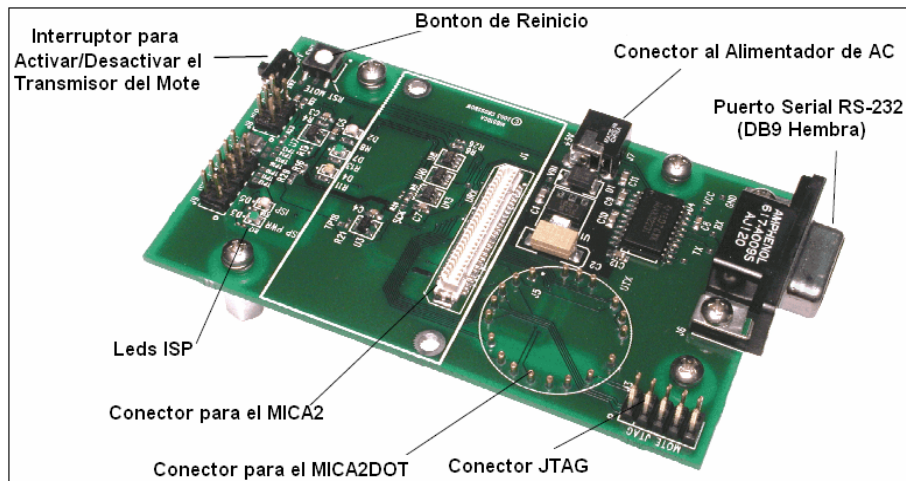


Figura 3.4: MIB510CA.

Esta plataforma cuenta con un procesador ATmega16L, denominado ISP (in-system processor), que lleva acabo la programación de los motes. El código se carga al ISP a través del puerto serial RS-232 y posteriormente éste programa el código en el mote.

Para programar las plataformas Mica2 y Mica2DOT, éstas deben conectarse a la plataforma MIB510 mediante los conectores Hirose de 51 y 19 pins respectivamente; y siempre programarse con el interruptor apagado (en el caso del mote Mica2) o sin pila (en el caso del mote Mica2DOT), con el fin de evitar una sobrecarga de voltaje.

El ISP también monitorea el voltaje de la plataforma (5 [v] usando el regulador o 3 [v] utilizando las baterías de los motes) y deshabilita la programación si el voltaje esta por debajo del límite requerido.

El procesador ISP está conectado a dos leds, uno verde etiquetado como “SP PWR” y uno rojo etiquetado como “ISP”. El SP PWR se usa para indicar el estado de la fuente de voltaje y el ISP encendido indica que la plataforma MIB510 tiene el control del puerto serial.

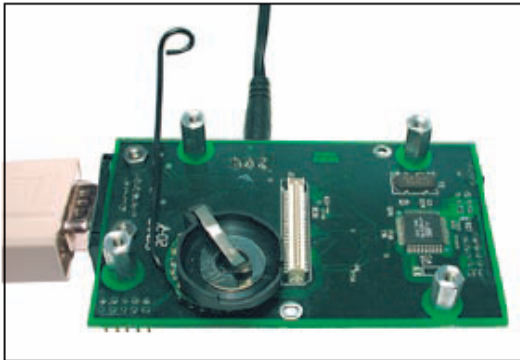


Figura 3.5: MIB510CA con Mica2DOT.

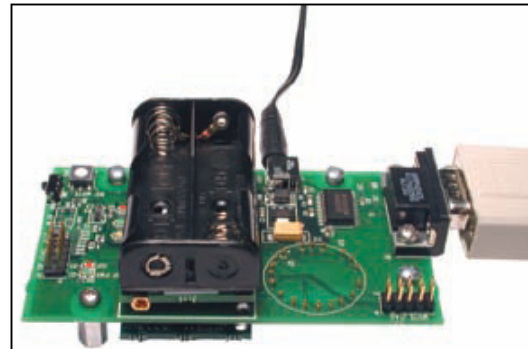


Figura 3.6: MIB510CA con Mica2

Como puede observarse en la Figura 3.6, la plataforma cuenta con otro conector Hirose de 51 pins, ubicado por debajo de la tarjeta, que permite la conexión simultánea de un sensor MTS310CA.

➤ **Mote Mica2 (MPR410CB)**

Esta plataforma cuenta con un microcontrolador ATmega128L, que descarga y ejecuta las aplicaciones en su memoria interna con ayuda de la puerta de enlace MIB510. El proceso de programación es supervisado mediante tres leds de colores rojo, verde y amarillo; los primeros dos reflejan los leds “ISP” y “SP PWR” de la plataforma MIB510.

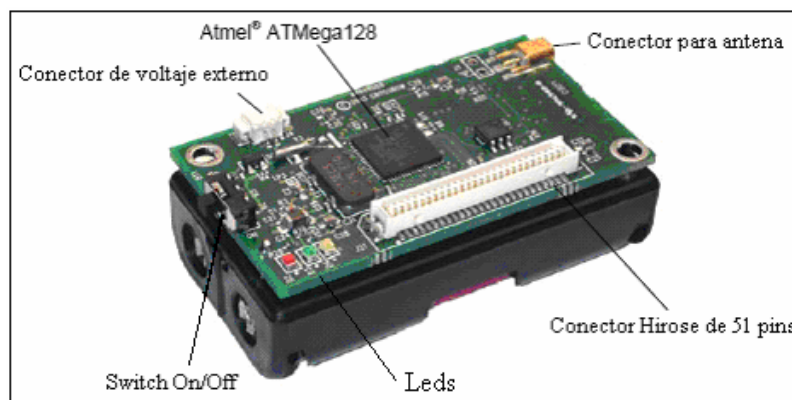


Figura 3.7: Mote Mica2.

El ATmega128L es un microcontrolador de 8 bits con memoria flash de 128 [kB] y procesa a una velocidad de 4 [MHz]. Consume únicamente 8 [mA] en estado de procesamiento y sólo 15 [μ A] en modo “dormido”. Éste bajo consumo de energía permite que el mote trabaje por periodos mayores a un año con un par de pilas AA. Para ello debe escribirse un código que mantenga al microcontrolador “dormido” la mayor parte del tiempo y así aumentar considerablemente la vida útil de las baterías.

El mote Mica2 está provisto de una memoria flash de 512 [kB] para almacenar los datos adquiridos. También cuentan con un convertidor A/D de 10 bits para que los datos del sensor puedan ser digitalizados. Las tarjetas de sensores MTS pueden adaptarse al mote mediante el conector Hirose de 51 pins.

Los componentes finales del mote MPR410CB son un radio y una antena monopolo. Éstos reciben y transmiten las señales de radio.

El radio es un transceptor RF CC1000 de la marca CHIPCON¹ que realiza la operación de transmisión y recepción de datos en la frecuencia de RF. Tiene una velocidad de transmisión aproximada de 40,000 [bps], está especialmente diseñado para sistemas que utilizan modulación FSK y opera en el rango de frecuencias de ISM (Industrial, Scientific and Medical) y SRD (Short Range Device). Cabe mencionar que debido a que el radio tiene un receptor muy sensible, los motes deben colocarse a una distancia mínima de 60 [cm].

Mientras se encuentra apagado, el radio consume menos de 1 [μ A], cuando está recibiendo datos consume 10 [mA] y durante el proceso de transmisión consume 25 [mA]. Además es compatible con los motes MPR4x0 y MPR5x0 (la letra “x” determina el modelo y la banda de frecuencia utilizada), y cuenta con la facilidad de ajustar la potencia de salida (de -20 a 5 [dBm]).

La antena empleada es una antena monopolo especialmente diseñada para la frecuencia de interés (433 [MHz])². Dicha antena no es más que un cable recubierto de un material aislante con una longitud igual a un cuarto de longitud de onda.

La longitud física de la antena es de aproximadamente 17.5 [cm]. Este tipo de antenas son de uso común en redes de sensores dado que son dispositivos pequeños y flexibles. Comúnmente se les conoce como antenas de látigo.

La Tabla 3.1, obtenida del manual de operación “MPR-MIB Series User Manual” de Cross-Bow, muestra el consumo de corriente típico para el radio CHIPCON CC1000 a determinada potencia. Además se especifica el número en hexadecimal que ajusta el radio a la potencia requerida.

¹ Compañía incorporada a Texas Instrument (TI)*, líder en el diseño de dispositivos transceptores de RF de corto alcance y baja potencia.

* TI es una empresa norteamericana que desarrolla y comercializa semiconductores y tecnología para computadoras. Es el tercer fabricante de semiconductores del mundo tras Intel y Samsung y es el mayor suministrador de circuitos integrados.

² El mote en cuestión (MPR410CB), utiliza cuatro canales en la banda de 433 [MHz] (433.05-434.79 [MHz])

Pout [dBm]	PA_POW (hex)	Consumo de Corriente Típico [mA]
-20	0x01	5.3
-19	0x01	6.9
-18	0x02	7.1
-17	0x02	7.1
-16	0x02	7.1
-15	0x03	7.4
-14	0x03	7.4
-13	0x03	7.4
-12	0x04	7.6
-11	0x04	7.6
-10	0x05	7.9
-9	0x05	7.9
-8	0x06	8.2
-7	0x07	8.4
-6	0x08	8.7
-5	0x09	8.9
-4	0x0a	9.4
-3	0x0b	9.6
-2	0x0c	9.7
-1	0x0e	10.2
0	0x0f	10.4
1	0x40	11.8
2	0x50	12.8
3	0x50	12.8
4	0x60	13.8
5	0x70	14.8
6	0x80	15.8
7	0x90	16.8
8	0xc0	20.0
9	0xe0	22.1
10	0xff	26.7

Tabla 3.1: Relación entre el Consumo de Corriente y la Potencia de Salida en un radio CHIPCON CC1000.

➤ **Mote Mica2DOT (MPR500)**

El mote Mica2Dot está diseñado para aplicaciones en donde el tamaño del dispositivo es importante, ya que tiene un diámetro aproximado de 3 [cm]. Al igual que los motes Mica2, cuenta con un microcontrolador ATmega128L que permite la descarga y ejecución de aplicaciones, y un transceptor RF CC1000 que realiza la transmisión y recepción de datos.

El radio opera también en la banda de 433 MHz y utiliza igualmente una antena monopolo para la radiación de las señales.

En esta plataforma sólo contamos con un led de color rojo, y una antena, que a diferencia del mote Mica2, se encuentra soldada a la placa.

Este mote cuenta además con un termistor, el cual es un componente superficialmente montado y se encuentra etiquetado como "RT1". La señal de salida está en ADC1 y es activada ajustando los pines PW6 y PW7 del procesador a "LO" y "HI" respectivamente.

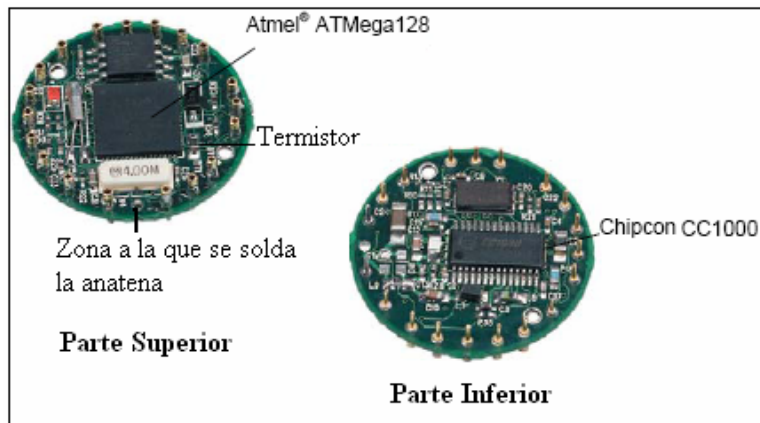


Figura 3.8: Mote Mica2Dot.

➤ Tarjeta de Sensores MTS310CA

La tarjeta de sensores MTS310 tiene múltiples modalidades de sensado. Estas modalidades incluyen: acelerómetro y magnetómetro de dos ejes, sensores de luz, temperatura, sonidos acústicos y un timbre resonador. Es compatible con los motes Mica2 y MicaZ. Puede ser usada en diversas aplicaciones; como detección de anomalías magnéticas o de vibración, localización, rastreo acústico; en robots, en redes de sensores, etcétera.

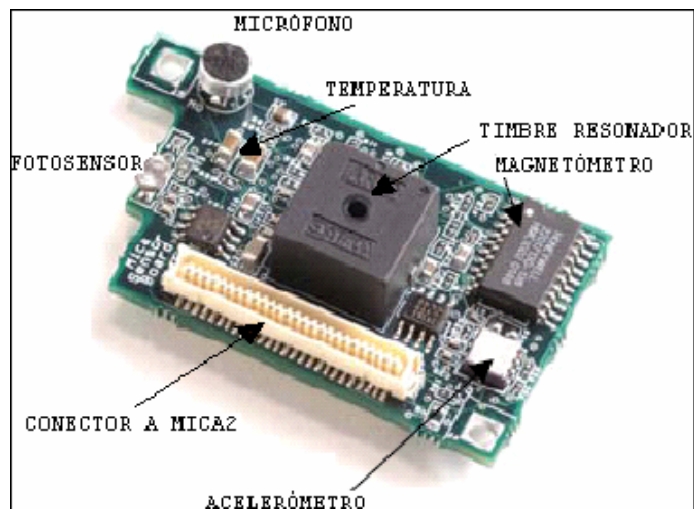


Figura 3.9: MTS310CA.

A continuación se describen algunas de las características de los sensores que fueron utilizados para este proyecto.

Luz y Temperatura

Los sensores de luz y temperatura comparten el mismo canal convertidor (ADC1), por lo que solamente debe encenderse un sensor a la vez o la lectura en ADC1 será errónea o sin sentido.

El MTS310CA tiene un sensor de luz y un termistor. El sensor de luz no es más que una foto celda CdSe. La sensibilidad máxima de la fotocelda es de un rayo de luz con una longitud de onda de 690 [nm]. La resistencia típica que presenta en presencia de luz es de 2 [k Ω], mientras que en la oscuridad es de 520 [K Ω].

Para utilizar el sensor de luz, la señal de control digital PW1 debe estar encendida, mientras que la salida del sensor debe conectarse al canal 1 del convertidor analógico-digital (ADC1). Cuando existe luz, la salida del circuito nominal es cercana a VCC o alcanza la escala completa, y en ausencia de ella, es decir, cuando esta oscuro, la salida del circuito nominal es cercana a tierra o a cero. Para encender el sensor de luz debe activarse la señal INT1.

El termistor es un componente montado en la localización RT2. Está configurado en un circuito divisor de voltaje con una lectura nominal media de 25 [°C].

La salida del circuito sensor de temperatura se encuentra disponible en ADC1. Para encender el sensor de temperatura debe activarse la señal INT2.

➤ **Tarjeta de Sensores MTS510CA**

La tarjeta de sensores MTS510 tiene las siguientes modalidades de sensado: un acelerómetro de dos ejes, sensor de luz y sensor de sonidos acústicos. Compatible únicamente con el mote Mica2Dot.

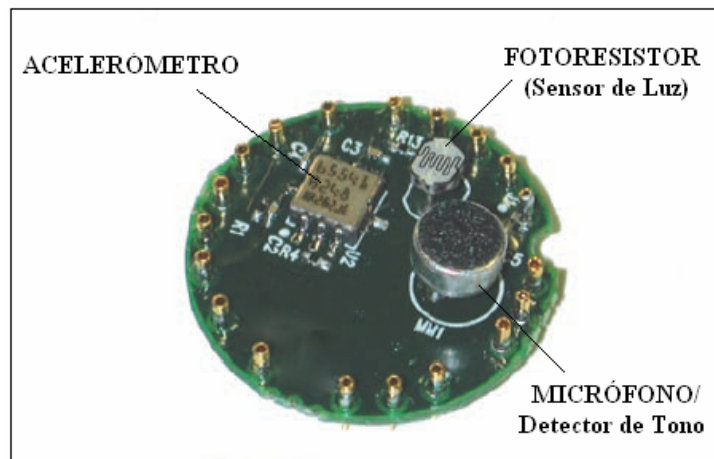


Figura 3.10: MTS510CA.

A continuación se describen algunas de las características de los sensores que fueron utilizados para este proyecto.

Luz

El sensor de luz no es más que una foto celda CdSe. La sensibilidad máxima de la fotocelda es de un rayo de luz con una longitud de onda de 690 [nm]. La resistencia típica que presenta en presencia de luz es de 2 [kΩ], mientras que en la oscuridad es de 520 [KΩ].

Para utilizar el sensor de luz, la señal de control digital PW1 debe estar encendida, mientras que la salida del sensor debe encontrarse conectada al canal 7 del convertidor analógico-digital (ADC7). Cuando existe luz, la salida del circuito nominal es cercana a VCC o alcanza la escala completa, y en ausencia de ella, es decir, cuando esta oscuro, la salida del circuito nominal es cercana a tierra o a cero.

3.2.2. SISTEMA OPERATIVO TinyOS

TinyOS es un sistema operativo basado en eventos, es decir, el software responde ante eventos que son propagados desde el nivel hardware hasta los niveles superiores. Se distingue entre otros sistemas operativos, en que permite a las aplicaciones manejar el hardware directamente, es decir, se compila junto con la aplicación incluyendo sólo los módulos requeridos por la aplicación, evitando así agregar funcionalidades innecesarias al sistema operativo. Debido a que está diseñado para ejecutarse en una WSN, utiliza muy poca memoria, maneja múltiples tareas y ayuda a optimizar el uso de baterías maximizando su tiempo de vida. Actualmente es el más utilizado en este tipo de sistemas ya que es de distribución libre.

Para implementar aplicaciones en TinyOS se utiliza una extensión del lenguaje C denominado nesC. Los programas en nesC están hechos con componentes ligados entre sí mediante interfaces claramente definidas; éstas permiten a cada componente ofrecer servicios a otros bien utilizar los servicios de aquellos. De manera que TinyOS no es más que un conjunto de componentes predefinidos que permiten manipular el hardware. Y, los servicios que estos componentes ofrecen a través de sus interfaces, pueden ser utilizados por cualquier otro conjunto de componentes creados por cualquier programador.

El lenguaje de programación nesC permite el desarrollo de aplicaciones compactas que necesitan poco espacio en memoria, lo cual es muy importante en los sistemas embebidos. En estos sistemas el diseño debe estar basado en un núcleo multi-enlaces que sea simple y eficiente.

Se puede entender a TinyOS como una pirámide en la que en el nivel más bajo se localizan los componentes del sistema, y en capas superiores, los componentes de uso habitual que llaman de forma implícita y encapsulada a los componentes del sistema. Para entender TinyOS, debemos tener presente tres clases de abstracción que son la base para su entendimiento y posterior programación:

- **Comandos.** Son las llamadas hacia abajo. Al llamar a un comando, éste dentro de su componente llama a otros comandos que pertenecen a componentes de capas inferiores.
- **Eventos.** Realizan un efecto inverso, una llamada hacia arriba. Un componente de bajo nivel avisa a uno de alto nivel que ha sucedido algo. Si sucede un evento, éste tiene mayor prioridad que cualquier tarea y se ejecuta. En caso de haber una interrupción dentro de una interrupción, lo más probable es que no sea ejecutada hasta que acabe la anterior, siendo esto algo crítico.
- **Tareas.** Son porciones de código que se ejecutan de forma asíncrona siempre y cuando el procesador no tenga que ejecutar ningún evento. Éstas se ejecutan por orden de llamada (FIFO) y en caso de que la cola esté vacía el procesador entrará en *standby* hasta que un nuevo evento lo despierte.

Muchos de los programas a los que llamamos mediante tareas tienen un evento indicando que su cometido ha sido realizado. Generalmente este evento puede desencadenar otra parte del código. Otros programas son inicializados y se activan cada vez que se realiza algo para lo que han sido programados. Muchas operaciones de bajo nivel son gestionadas por el sistema operativo y esto nos permite centrarnos en la aplicación.

Cuando programamos y compilamos una aplicación, ésta va siempre unida de manera implícita al sistema operativo, de forma que no existe el sistema operativo (previamente cargado) ni el programa compilado, sino el conjunto que es subido al sistema empujado cada vez que se requiere actualizar el programa.

➤ **NesC**

En nesC cada aplicación está construida a base de componentes enlazados para formar un ejecutable. Los componentes pueden proporcionar o utilizar interfases de otros componentes. La única forma de acceder al uso de estos componentes es mediante estas interfases que son bidireccionales, es decir, mediante eventos y mediante comandos. Los comandos deben ser implementados por el proveedor de la interfaz, mientras que los eventos deben ser implementados por el usuario de la interfaz.

Implementación de los componentes

En nesC se definen dos clases de componentes: los módulos y las configuraciones. Los módulos contienen el código de la aplicación, implementando una o varias interfases. Por otra parte, las configuraciones enlazan las interfases de los componentes que se utilizarán en determinado módulo. Este fichero es crucial ya que si no está bien configurado la aplicación no compilará.

Tanto módulos como configuraciones e interfases, utilizan la misma extensión: *.nc, sin embargo se sigue un código de letras para diferenciar unos de otros. Por ejemplo, cuando trabajamos con un programa "ejemplo": ejemploM.nc sería el módulo, ejemploC.nc sería el fichero configuración y ejemplo.nc la interfase.

Modelo concurrente

TinyOS ejecuta únicamente un programa consistente en los componentes del sistema y en los componentes propios de la aplicación. Existen dos lazos de ejecución: las tareas y los eventos. Si recordamos, los eventos lanzaban la ejecución de un código vinculado al evento ocurrido y/o llamaban a comandos que se encargaban así mismo de tareas específicas vinculadas al evento.

El lenguaje nesC debe tener conocimiento de que comandos y eventos serán ejecutados como parte de un evento, para ello en su definición se les debe declarar con la palabra clave *async*.

Debido a que tareas y eventos pueden ser detenidos para permitir la ejecución de código asíncrono de mayor prioridad, los programas en nesC son susceptibles a padecer *data races*³. Las condiciones de aparición de *data races* pueden evitarse mediante el acceso a datos compartidos exclusivamente en el interior de las tareas o bien encapsulando todos los accesos dentro de secciones *atomic*. Las secciones *atomic* aseguran la ejecución de todo el código que encapsulan, si llega una interrupción, ésta esperará hasta que la ejecución abandone la sección *atomic* para ejecutarse. El compilador de nesC advierte de *data races* potenciales en tiempo de compilación para que el programador pueda rediseñar su código.

3.2.3. SOFTWARE DE APLICACIÓN

Los paquetes que se describen a continuación, así como cada una de las aplicaciones que serán mencionadas (XMesh, XSensor, Surge_Reliable, Surge_Reliable_Dot, etcétera), han sido desarrolladas por la compañía Cross-Bow sobre la plataforma TinyOS, son de uso libre y están disponibles en la página de dicha compañía (www.xbow.com).

SURGE-VIEW

Es un software implementado por Cross-Bow que recopila datos de la red y permiten observarlos de manera gráfica en la computadora.

Los motes Mica2 y Mica2Dot son programados con ayuda de la plataforma MIB510 desde una ventana de cygwin. El programa se carga a los motes desde los directorios `opt/tinyos-1.x/contrib/xbow/apps/Surge_Reliable` u `opt/tinyos-1.x/contrib/xbow/apps/Surge_Reliable_Dot`, según el mote que desee programarse; y se ejecuta desde una ventana de comandos de MS-DOS en el directorio `c:\Archivos de Programa\Surge-View`.

`Surge_Reliable` es un algoritmo para redes con topología en malla basado en el protocolo de enrutamiento XMesh (desarrollado por Alec Woo y Jason Hill en la universidad de Berkeley California).

³ Son situaciones que ocurren cuando diferentes procesos acceden a datos compartidos sin estar sincronizados. Pueden provocar comportamientos inesperados del programa por lo que es importante detectarlas a tiempo.

Al momento de ejecutarse, Surge-View despliega la siguiente información:

- **Mapa de conectividad.** Muestra la topología de la red.

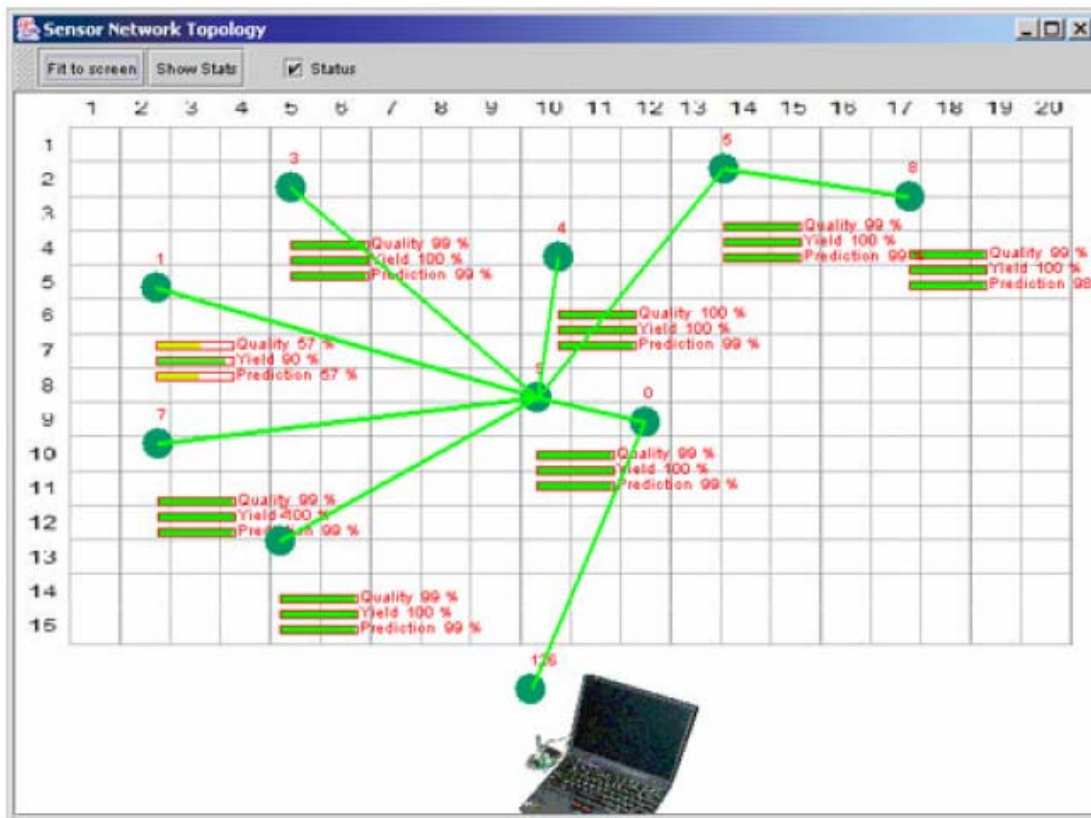


Figura 3.11: Surge-View : Topología de la Red.

- **Tabla de estadísticas.** Recopila diversa información de los nodos, como voltaje, paquetes enviados, paquetes recibidos, rendimiento, etcétera. (Figura 3.12), y
- **Recuadro de sincronía.** En él se observa el puerto a través del cual es transmitida la información hacia la computadora, así como la velocidad de transmisión y el número de paquetes recibidos (Figura 3.13).

Id	Rec.	Sent	Yield	Level	Duty...	Parent	Quality	Volta...	P1	P2	Min...
126											
9	335	323	1.037	1.009	1.081	0	0.949	3.165	0	2	5.835
8	310	318	0.975	2.026	2.934	6	0.996	3.045	2	6	5.337
7	337	318	1.06	1.214	1.747	0	0.82	3.089	0	9	5.686
6	531	322	1.649	1.066	1.309	0	0.98	3.119	0	2	5.827
5	314	320	0.981	1.987	1.161	6	0.969	3.157	2	6	5.867
4	332	322	1.031	0.976	1.249	0	0.961	2.967	0	9	5.263
3	304	320	0.95	1.924	1.45	6	0.996	3.089	9	0	5.659
2	323	321	1.006	1.006	1.448	0	0.502	3.089	0	4	5.82
1	329	319	1.031	1.964	1.192	4	0.996	3.134	2	4	4.984
0	329	329	1	0	100....	126	0	3.313	126	126	0

Figura 3.12: Surge-View : Tabla de estadísticas.

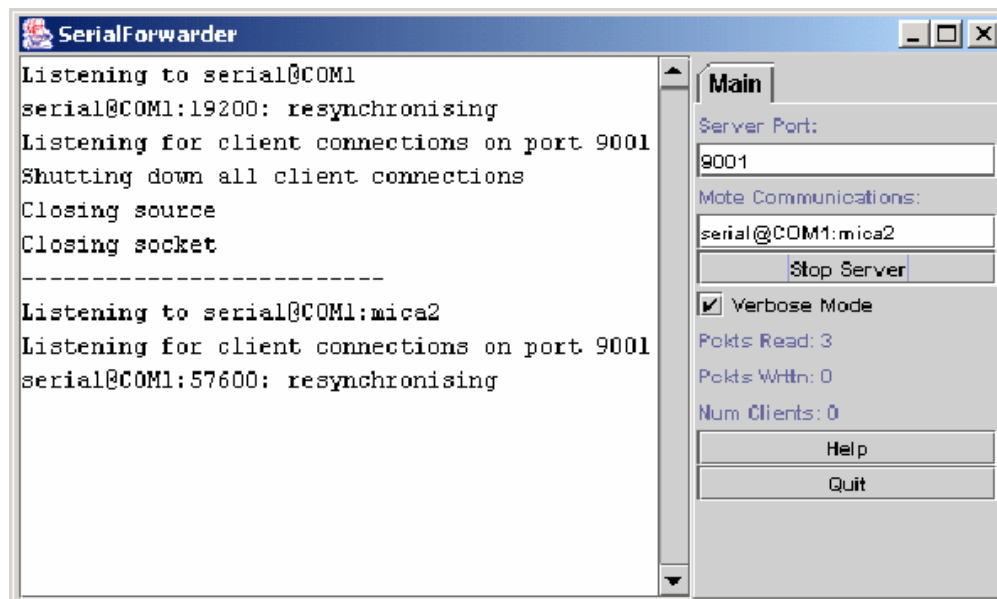


Figura 3.13: SerialForwarder: Muestra la sincronización de los puertos.

Este software se apoya en el programa SerialForwarder, el cual está escrito en Java y es utilizado para leer paquetes de datos desde un puerto serial y remitirlos a través de un puerto de conexión hasta el servidor. Incluye además, el programa HistoryViewer, que ejecuta topologías y tablas estadísticas previamente grabadas al servidor.

MOTE-VIEW

Este software es una interfase entre el usuario y la WSN implementada. Provee las herramientas necesarias para monitorear la red, ya que permite al sistema conectarse a una base de datos para analizar y graficar las lecturas obtenidas desde los nodos sensores. Algunas de las herramientas que nos proporciona son:

- Exportación de los datos a archivos xml o csv.
- Cambio de la frecuencia, potencia o canal de transmisión de los nodos.
- Definición de acciones de alerta de acuerdo a determinados umbrales de comportamiento.
- Observación los parámetros de sensado: luz, temperatura, presión, etcétera, en las unidades requeridas.
- Determinación del estado de los nodos basándose en los colores que muestran: *gris*: resultados no recibidos, *verde*: resultados frescos en los últimos 20 minutos, *verde claro*: resultados no recibidos por más de 20 minutos, *amarillo*: resultados no recibidos por más de 40 minutos, *naranja*: resultados no recibidos por más de 60 minutos, y *rojo*: resultados no recibidos por más de un día.
- Creación de gráficas de manera simultánea para un máximo de 24 sensores con 3 magnitudes de sensado diferentes por mote. El eje “y” en la gráfica representa el tiempo transcurrido, mientras que en el eje “x” se representan los datos sensados en unidades ingenieriles.
- Además nos permite programar, configurar, agregar y eliminar nodos de la red.

Algunas de estas funciones serán explicadas en capítulos posteriores, y se detallarán las acciones llevadas a cabo para el proceso de monitoreo de la red implementada.

Las herramientas antes mencionadas se ejecutan a través de la ventana de interfase de Mote-View, Figura 3.14. Esta figura corresponde a un demo de la aplicación Mote-View para un sensor MTS310; en la parte superior podemos observar las barras de menú y de herramientas, debajo de ellas se visualizan el nombre y el número que identifica a cada uno de los nodos que conforman la red, así como los parámetros sensados por éstos (voltaje, temperatura, luz, aceleración, etcétera). Finalmente, la última columna, muestra la hora en tiempo real.

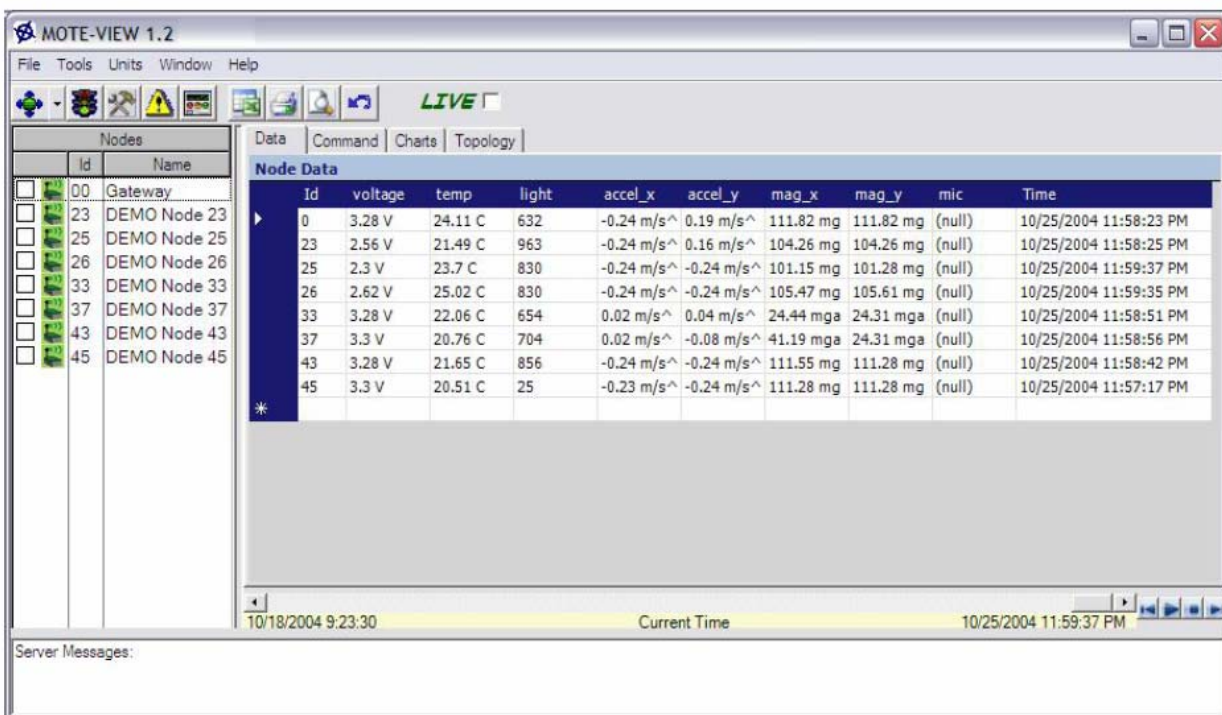


Figura 3.14: Ventana de interfase Mote-View.

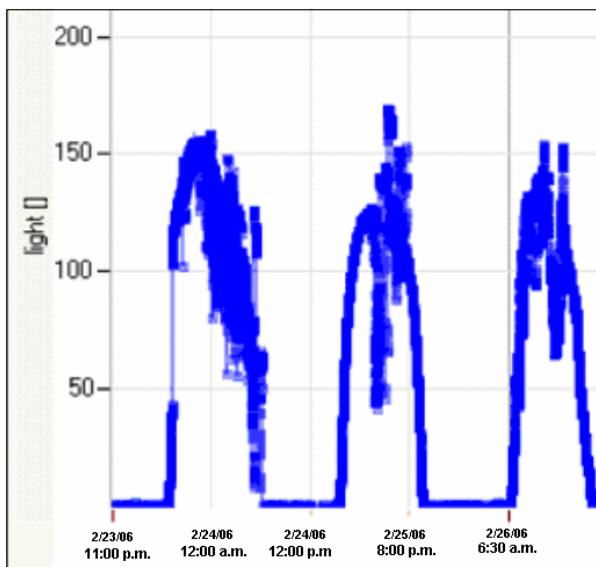
Capítulo 4

VALIDACIÓN DEL SISTEMA

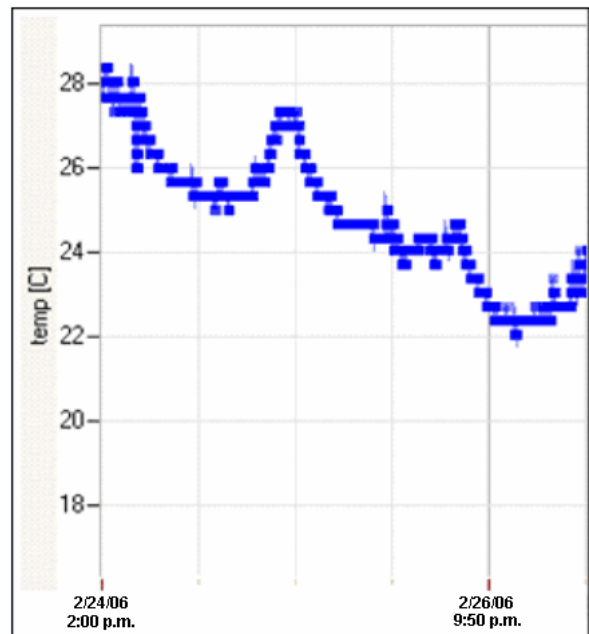
Antes que nada se llevaron a cabo pruebas para cerciorarnos de la viabilidad del proyecto, es decir, que fuese factible construir un sistema de monitoreo en base a los dispositivos, software y sensores de los que se disponía.

Aunque los dispositivos sensores MTS310 y MTS510 permiten grabar sonido no detectan señales de baja intensidad, de manera que no contábamos con un dispositivo que nos avisara de robo o intrusión a la oficina, puesto que un intruso siempre irrumpirá con cautela y haciendo el menor ruido posible. Otro problema derivado del uso de este dispositivo suponía que grabar los acontecimientos en una oficina violaría la intimidad de los usuarios. Debido a estos inconvenientes, se llegó a la conclusión de que el uso de dicho dispositivo no era viable para fines de vigilancia; y como tampoco disponíamos de sensores de movimiento, esta aplicación tuvo que descartarse.

Las pruebas con los sensores de luz y temperatura demostraron que dichos dispositivos sí nos permitirían cumplir con las necesidades de monitoreo.



Gráfica 4.1: Tiempo vs Luz.



Gráfica 4.2: Tiempo vs Temperatura

De las gráficas obtenidas puede deducirse que los dispositivos son sensibles a los cambios de luz y temperatura en su entorno. En la Gráfica 4.1 se observan claramente los cambios en el nivel de iluminación. Durante la noche, por ejemplo a las 11 pm, el nivel es de cero, mientras que en la madrugada (6 am), los niveles de luz comienzan a subir de manera gradual.

De la misma forma, en la Gráfica 4.2, puede observarse una temperatura cálida durante la tarde (28 [°C] a las 2:17 pm) y una temperatura templada durante la noche (22 [°C] a las 9:51 pm).

Estos resultados muestran que los valores registrados por los motes corresponden a valores estimados, lógicos y fehacientes para esas horas y condiciones del día.

Por otro lado, con el fin de satisfacer los requisitos técnicos y tecnológicos señalados en el Capítulo 2, se adoptaron las siguientes soluciones.

a) Estación central encargada de recibir las señales.

Como se mencionó en el Capítulo 3, la plataforma MIB510CA es el enlace que nos permite retransmitir los datos recibidos a una computadora local, ya que cualquier mote Mica2 puede funcionar como una estación base al ser interconectada a ésta.

El software de aplicación Surge y Mote-View fueron los medios para llevar a cabo el procesamiento y visualización de los datos adquiridos, así como la emisión de las alarmas correspondientes.

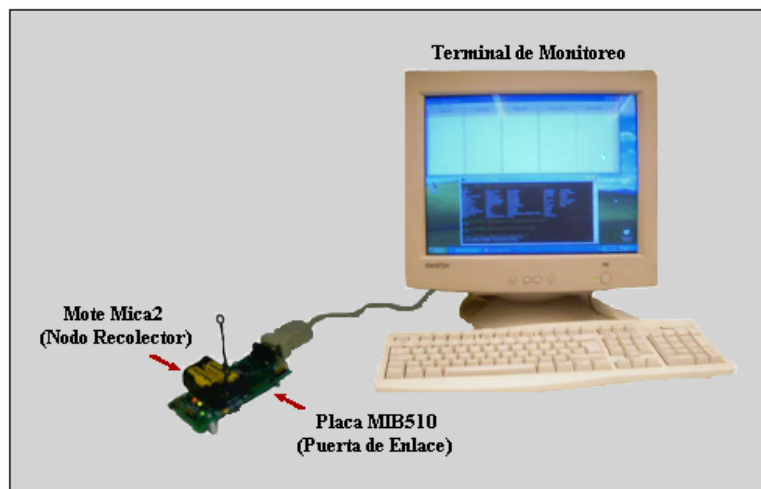


Figura 4.1: Estación de Monitoreo.

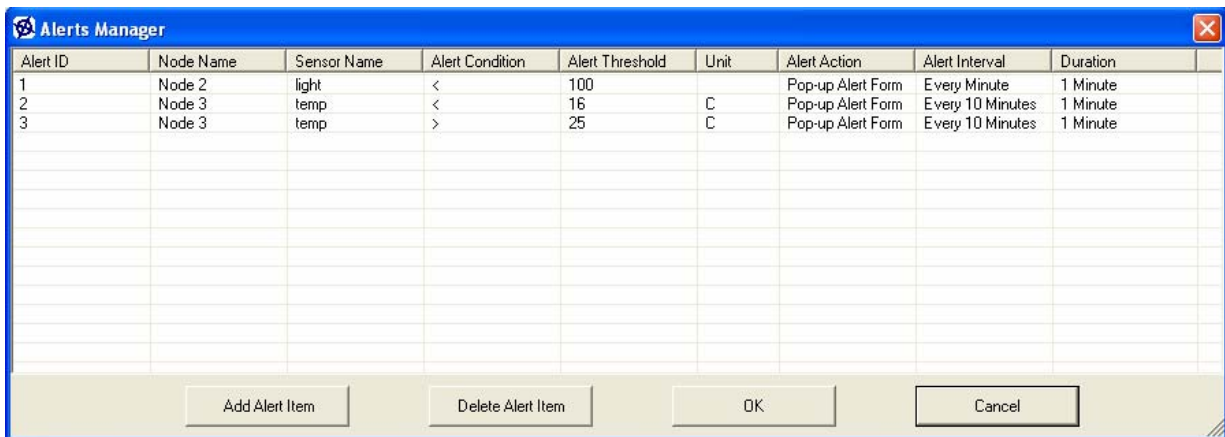
b) Rápida detección y resolución del problema

El software de aplicación Mote-View 1.2 cuenta con un gestor de alertas que permite al usuario definir una condición de alarma basándose en cualquier dato sensado desde cualquier nodo de la red. Una alerta es un evento programable que se ejecuta cuando los datos sensados exceden un umbral previamente definido. Existen dos posibles respuestas a una alerta:

- Desplegar una ventana de diálogo en el que se especifican el nodo, tiempo y condición en que se ha sobrepasado un umbral.
- Enviar un e-mail a una dirección previamente definida.

El gestor de alertas nos permite establecer un intervalo de alerta y un tiempo de duración. El intervalo de alerta es el tiempo durante el cual se lleva a cabo la acción, mientras que el tiempo de duración se refiere al periodo durante el cual debe cumplirse la condición de alerta antes de que la alerta sea activada.

Consideremos el ejemplo mostrado en la tabla de alertas de la Figura 4.2. Para el caso del nodo 3, debe rebasarse el umbral establecido (temperatura menor a 16 [°C] o mayor a 25 [°C]) durante el intervalo de un minuto, antes de que la alarma sea activada. Asimismo, la alerta se estará ejecutando cada diez minutos hasta que la condición establecida vuelva a su estado óptimo.



Alert ID	Node Name	Sensor Name	Alert Condition	Alert Threshold	Unit	Alert Action	Alert Interval	Duration
1	Node 2	light	<	100		Pop-up Alert Form	Every Minute	1 Minute
2	Node 3	temp	<	16	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
3	Node 3	temp	>	25	C	Pop-up Alert Form	Every 10 Minutes	1 Minute

Figura 4.2: Gestor de Alertas de Mote-View.

Con la ayuda de una compresa de agua fría se bajó la temperatura del sensor por debajo de los 25 [°C], posteriormente se dejó que éste fuese adquiriendo la temperatura ambiente en ese momento (29.3 [°C]), las alertas enviadas se muestran en la Figura 4.3.

En la figura se observa que después de la primera alerta los mensajes son enviados cada diez minutos mientras que la temperatura no vuelva a estar dentro del umbral establecido.

El gestor de alertas no permite establecer intervalos de alerta y de duración menores a un minuto, de manera que el tiempo de detección de la señal y el despliegue de la alerta no es nulo, es decir, el sistema no responde de manera inmediata. A pesar de este inconveniente, el sistema puede considerarse aceptable dado el entorno que estamos monitoreando.

En un sistema de monitoreo de oficina, un retraso de un minuto entre la detección y la resolución del problema no implica mayores riesgos, pues el usuario no experimentará un calor o un frío extremo, ni tampoco una oscuridad total, ya que las magnitudes consideradas en el umbral óptimo tienen una diferencia de 2 [°C] para la temperatura y un valor fijo para la luz. Sin embargo, en otro caso particular, por ejemplo si se

monitoreasen los signos vitales de un paciente, este retraso podría implicar graves consecuencias.

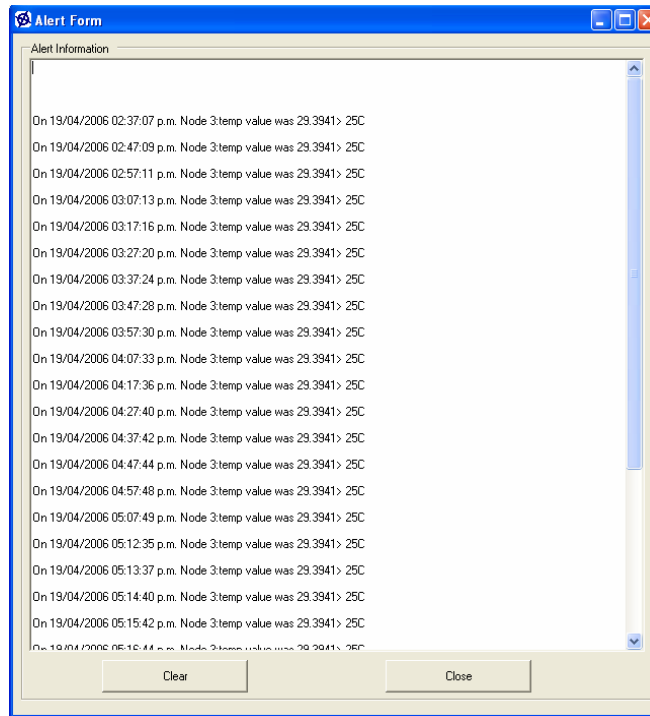


Figura 4.3: Recuadro de Alertas.

c) Retraso de los mensajes

Para determinar el retraso que sufren los mensajes en la red, se realizaron pruebas desde diversos nodos, la información fue recabada a través de la tabla de estadísticas de Surge-View (Figura 4.4).

Id	Rec.	Sent	Yield	Level	Duty Cycle	Parent	Quality	Voltage	P1	P2	Min Cut
7	115	115	1	1.913	0	4	0.859	2.9	4	4	1.945
6	116	115	1.009	1.931	0	4	0.929	3.11	4	4	1.945
126											
5	115	115	1	2	0	4	0.984	2.7	4	4	2.882
4	115	115	1	0.965	0	0	0.996	3.1	0	0	1.945
3	7	7	1	1.286	0	7	0.816	2.9	7	7	0.816
2	114	115	0.991	0.974	0	0	0.949	2.762	0	0	1.945
0	115	115	1	0	0	126	0	3.119	126	126	0

Figura 4.4: Tabla de Estadísticas.

Esta tabla proporciona una información sobre las condiciones de asignación de rutas dentro de la red. La columna *Id* nos sirve para identificar el nodo que se está evaluando y de cada nodo podemos observar además; el número de paquetes recibidos, número de paquetes enviados, rendimiento de la red, calidad del enlace, etcétera.

Como puede observarse en la tabla, no todos los nodos sufren retraso en sus mensajes, en este caso sólo se observan pérdidas de paquetes en los nodos 2 y 6. Sin embargo, el rendimiento de éstos nodos no se ve afectado en menos de un 90%, lo cual implica que el sistema de monitoreo es viable, ya que la pérdida de paquetes que se observa no representa un deterioro en el funcionamiento de la red.

La ubicación de los nodos sensores durante la toma de datos se muestra en la Figura 4.5.

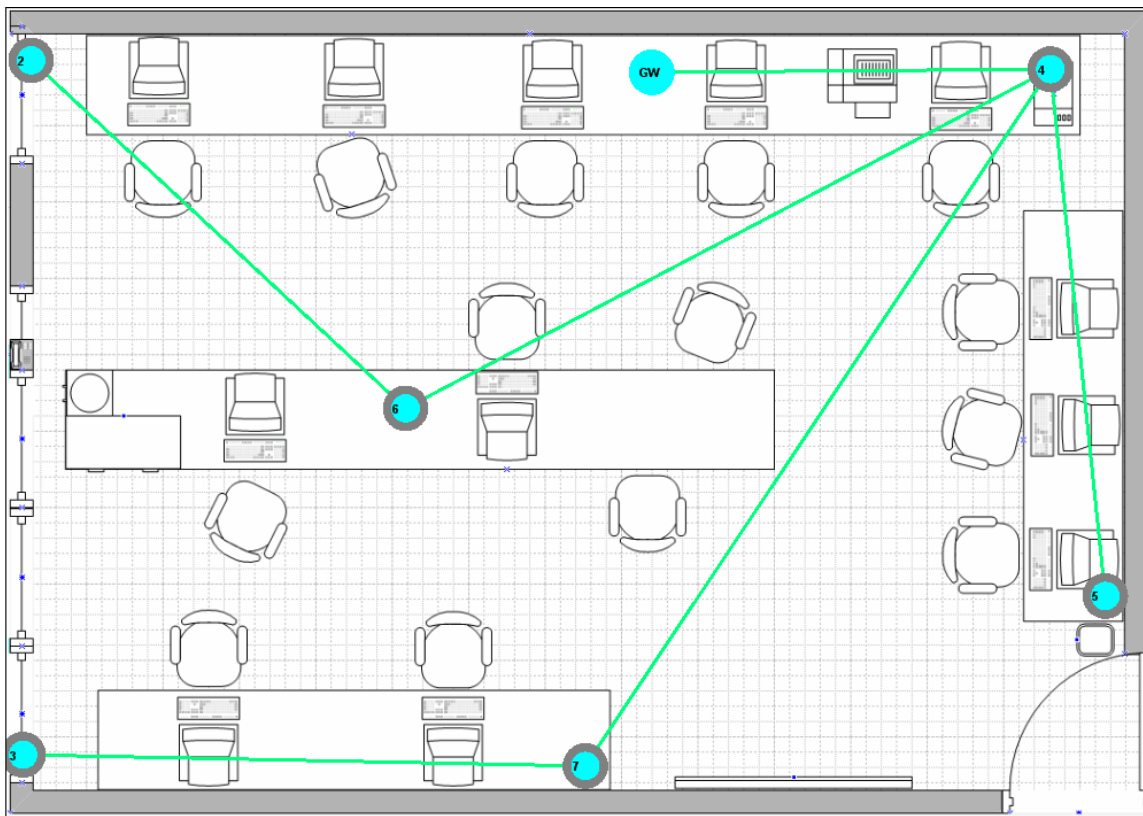


Figura 4.5: Topología de la Red para pruebas de envío de mensajes.

d) Alcance de transmisión

El alcance de la señal de RF tiene mucho que ver con la potencia de transmisión y con la disposición de los sensores dentro del entorno.

Para verificar que los nodos del sistema tuviesen un alcance mayor o igual a 5 [m], se tomaron muestras de la calidad y rendimiento de un enlace a determinadas distancias y a diferentes niveles de potencia. Estos datos se recabaron a través de la tabla de estadísticas del software Surge-View.

Para este primer análisis se recabó la información de un sólo nodo (Tabla 4.3), ya que la finalidad era determinar las condiciones óptimas en términos de potencia y distancia a las que debían fijarse los demás nodos.

Distancia [m]	DATOS DEL ENLACE [%]										
	-20 dBm		-3 dBm		-10 dBm		0 dBm		10 dBm		
	Q	Y	Q	Y	Q	Y	Q	Y	Q	Y	
1	89	95	100	94	100	100	100	100	100	100	100
2	74	46	100	95	100	100	100	100	100	100	100
3	---	---	95	96	98	100	99	100	100	100	100
4	---	---	90	96	99	100	99	100	100	100	100
5	---	---	97	100	95	100	99	100	100	100	100
6	---	---	86	100	93	97	95	100	100	100	100
7	---	---	86	100	87	93	93	100	100	100	100

Tabla 4.3: Alcance y Calidad de transmisión.

Q: Calidad del Enlace; Y: Rendimiento de la Red.

En base a los resultados obtenidos se resolvió programar los sensores a una potencia de transmisión de -3 [dBm], y colocarlos a distancias menores o iguales a 5 [m]. Dichas condiciones proporcionan el consumo de corriente mínimo dentro del cual se conserva un grado de rendimiento y de calidad de la señal aceptables para nuestra aplicación.

Programar los sensores a una potencia de transmisión de -20 [dBm] reduciría el consumo de corriente de los nodos sensores en un 55[%] (ver Tabla 3.1), además de que haría al sistema menos propenso a interferencias. Sin embargo, se necesitaría un mayor número de nodos para cubrir el área de monitoreo, ya que la distancia máxima entre ellos debería ser de 2 [m], aumentando de esta manera, el costo de la red.

Cobertura

Otro factor que juega un papel fundamental en la transmisión de la señal de RF hasta la estación base, es la cobertura de la señal enviada. Dicha cobertura, así como el rango y rendimiento de la señal, no sólo se ven afectados por la forma en que se dispongan los sensores, sino que además dependen de una antena que proporcione la cobertura adecuada para el entorno escogido.

Tal como se mencionó en el Capítulo 3, los motes utilizados en este proyecto emplean una antena monopolo. Este tipo de antenas tienen en teoría una radiación isotrópica, es decir, transmiten energía uniformemente en todas direcciones.

La siguiente gráfica muestra el patrón de radiación obtenido para la antena de uno de los motes utilizados en este proyecto.

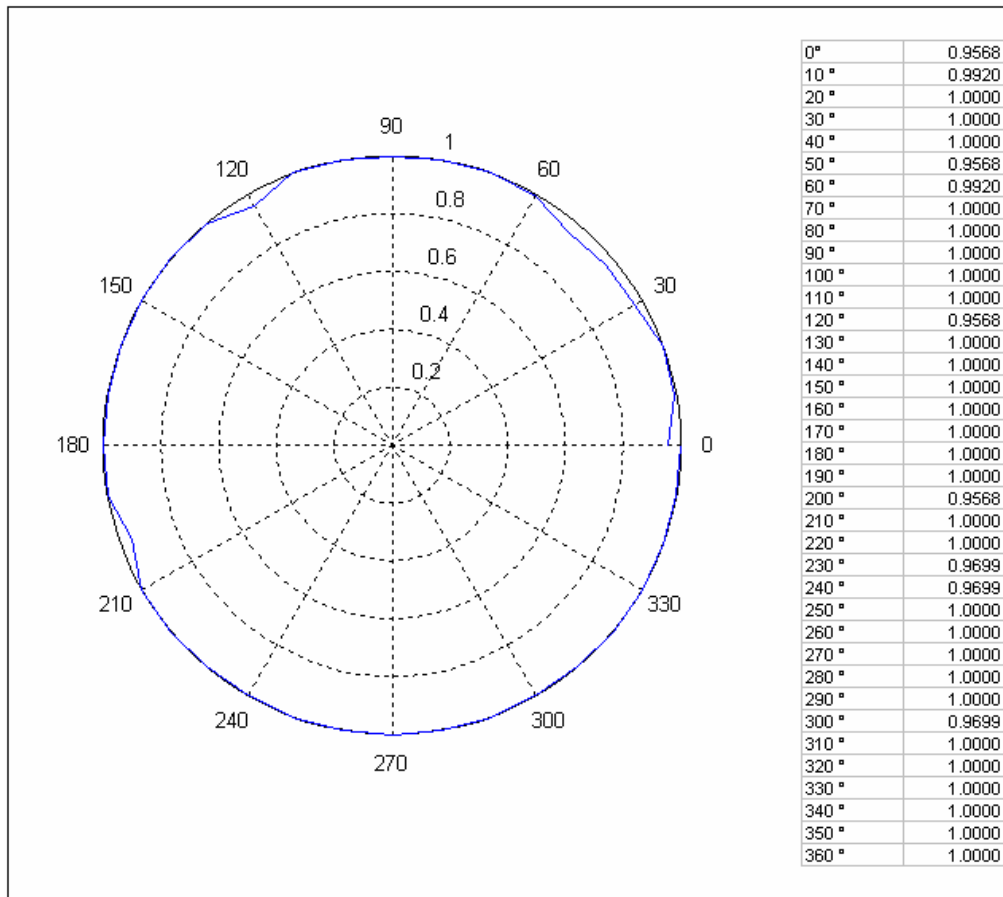


Figura 4.5: Patrón de Radiación de una Antena Monopolo.

Como puede observarse, aunque el patrón de radiación no es estrictamente isotrópico, ya que en algunos puntos la intensidad de radiación no es la misma, las características de transmisión de energía cumplen satisfactoriamente con los requerimientos para el tipo de red a implementar en este proyecto (una red de topología en malla).

La señal es emitida en todas direcciones, de manera que el nodo más cercano puede recibir la señal y de igual manera radiarla hasta su vecino más próximo.

Polarización

Otra característica a tomar en cuenta para una buena transmisión y recepción de una señal de RF, es la polarización de la antena utilizada.

La polarización es la dirección de la componente de campo eléctrico de una onda electromagnética. Una antena horizontal al plano de tierra genera una polarización horizontal; mientras que una antena perpendicular al plano de tierra origina una polarización vertical.

Para tener una máxima respuesta de transmisión y recepción, las antenas deben estar orientadas con la misma polarización (Figura 4.6).

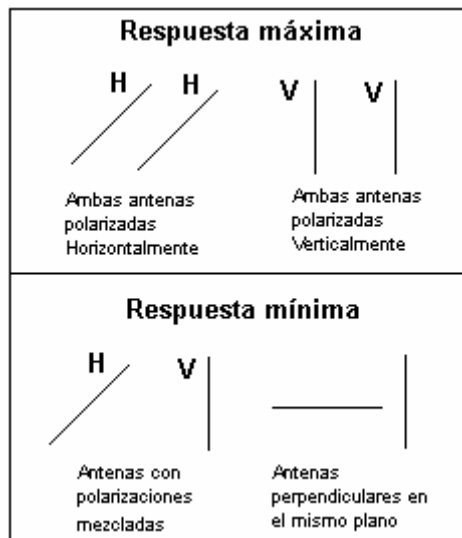


Figura 4.6: Efectos de Polarización.

De manera que debe tenerse especial cuidado en la disposición de las antenas de cada uno de los motes. En la implementación se procuró que las antenas de los motes se mantuviesen en una posición vertical.

e) Calidad de la señal

Como se mencionó en el Capítulo 3, el radio que utilizan los motes Mica2 y Mica2Dot, es un transceptor RF CC1000 especialmente diseñado para sistemas que utilizan modulación FSK y opera en el rango de frecuencias de ISM.

Se realizaron algunas pruebas para determinar si alguna interferencia afectaba la calidad de nuestras señales.

La siguiente tabla muestra la calidad del enlace bajo ciertas condiciones.

Condición <i>A través de:</i>	Calidad del Enlace [%]	Rendimiento [%]
Pared Falsa	100	100
Pared de Concreto	100	100
Mueble de madera	100	100
Cinco gabinetes	100	92
Dos oficinas	90	90

Tabla 4.4: Calidad de la Señal.

De la Tabla 4.4 podemos concluir que los dispositivos sensores satisfacen las necesidades del sistema a implementar, ya que el nivel de calidad de la señal se mantiene dentro de un rango aceptable a pesar de los obstáculos localizados en el entorno de monitoreo. Esto también puede concluirse de la Tabla 4.3, cuyos datos se obtuvieron dentro del laboratorio de pruebas y en los cuales no se observa degradación de la señal debido a interferencias externas.

Cabe mencionar que las condiciones de recepción, transmisión y calidad de la señal, dependen en gran manera de la estructura del laboratorio de pruebas. La disposición de los sensores y las condiciones del entorno de monitoreo pueden provocar que las señales reboten en las paredes, ventanas, muebles, equipos de cómputo, e inclusive en personas; de manera que la señal llega por diferentes vías al nodo destino (fenómeno conocido como efecto multitrayectoria).

Si la onda reflejada viaja una mayor distancia para llegar al nodo destino, entonces ocurrirá una diferencia de fase entre la onda directa y la onda reflejada. Esta diferencia de trayectoria puede ocasionar que las ondas se sumen o se cancelen al llegar al nodo destino; efecto que ocasionará pérdidas en la señal enviada.

De acuerdo a las pruebas realizadas en el entorno de monitoreo (Figura 4.4 y Tablas 4.3 y 4.4), las señales conservan una calidad y rendimiento adecuados para nuestros propósitos de monitoreo.

f) Fiabilidad de la Red

Las Redes de Sensores Inalámbricas poseen un protocolo de enrutamiento encargado de la conectividad entre nodos, es decir, del enrutamiento de la información a través de los diversos nodos de la red.

Los protocolos de enrutamiento en redes Ad-Hoc se pueden dividir en dos grupos: basados en tablas de enrutamiento y basados en enrutamiento bajo demanda (también conocidos como protocolos *preactivos* y protocolos *reactivos*). Los protocolos preactivos mantienen sus rutas actualizadas y consistentes en todo momento. Cada nodo mantiene una o más tablas con los datos para encaminarse hacia cualquier otro nodo de la red. Este protocolo tiene como ventaja su baja latencia y como inconveniente una alta sobrecarga de tráfico en la red.

Los protocolos *reactivos* se basan en calcular la ruta óptima hacia un determinado destino solamente cuando es necesario. Cuando un nodo requiere una ruta hacia un destino concreto se inicia un proceso de descubrimiento de ruta. Este proceso termina cuando se encuentra un camino hacia el destino o cuando se examinan todas las alternativas y ninguna lleva al destino final. Cuando la ruta es descubierta, es necesario mantenerla hasta que el destino se vuelva inalcanzable o la ruta deje de ser necesaria. Estos tipos de protocolo intentan reducir la sobrecarga generada por los mensajes de actualización de rutas periódicas de los protocolos proactivos, el inconveniente que tienen, es el retardo inicial que puede ocasionar el envío de mensajes por una nueva ruta.

En esta aplicación los sensores se programaron con Surge_Reliable. Dicho algoritmo, como se mencionó con anterioridad, está basado en el protocolo XMesh, creado por Alec Woo y Jason Hill en la universidad de Berkely California, y catalogado como protocolo preactivo.

El protocolo XMesh es una tecnología de arquitectura abierta y flexible para sistemas de redes inalámbricas cuya plataforma de control está diseñada para soportar aplicaciones TinyOS.

El software XMesh se ejecuta en cada uno de los nodos o motes dentro de la red inalámbrica. Usando XMesh, cada mote forma de manera dinámica una red en malla confiable utilizando técnicas de enrutamiento para redes Ad-Hoc. Algunas de las características ofrecidas por XMesh son:

1. *True mesh*. Software que permite a la red autoconfigurarse y repararse ante el fallo de alguno de sus nodos.
2. *Extensión de cobertura*. El área de cobertura se amplía con la adición de nuevos motes.
3. *Estado de alerta con bajo consumo de energía*. Los motes se activan varias veces por segundo para verificar si existe algún dato listo para ser transmitido.

En este tipo de configuración, los motes pueden formar su propia área de cobertura a pesar de estar fuera del rango de la estación base.

El tipo de protocolo de enrutamiento utilizado por Surge-View se denomina *Multi-Hop*, el cual es una funcionalidad proporcionada por TyniOS. En los Apéndice A y B podemos encontrar una pequeña descripción del funcionamiento de este programa y el código principal de *Surge-View* (programa instalado en los motes).

Podemos deducir de lo anteriormente expuesto, que el protocolo de enrutamiento funcionará en un plano invisible al usuario y será el encargado del buen funcionamiento de la red ante el fallo de alguno de sus nodos y/o enlaces.

g) Protección de los nodos sensores

Los nodos sensores cuentan con una caja protectora que los resguarda de condiciones climáticas y caídas, entre otras cosas. Esta caja permite que los

dispositivos del nodo sensor se mantengan en buen estado, además de evitar costos de mantenimiento. Por otro lado, la carga de las baterías se monitorea a través del software Mote-View, de manera que éstas puedan ser sustituidas antes de que el nodo deje de funcionar.

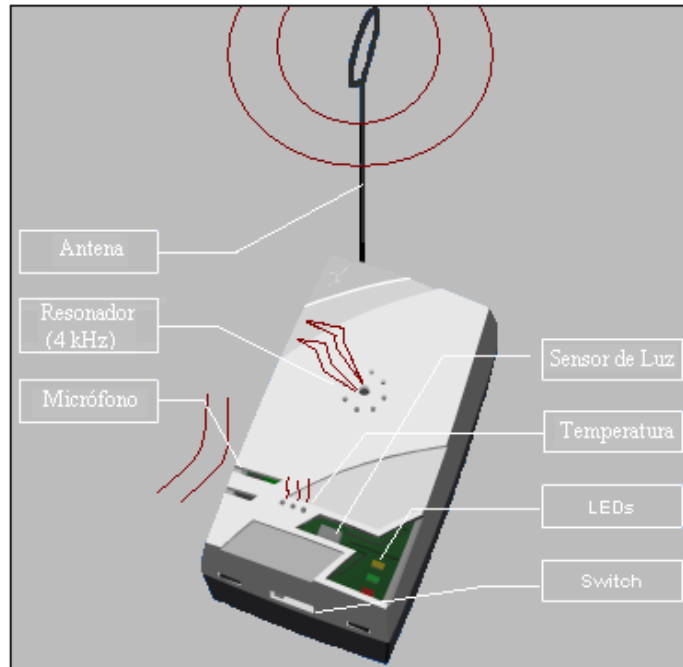


Figura 4.7: Caja protectora para los nodos sensores.

h) Manejo de la Red

Surge-View y Mote-View son programas de aplicación que proveen al usuario de una interfase muy amigable.

Estas aplicaciones muestran una topología de la red y es posible extraer cualquier dato estadístico requerido. También permiten establecer cambios de manera sencilla, como por ejemplo la eliminación, configuración o programación de nuevos nodos o tareas.

Capítulo 5

IMPLEMENTACIÓN DEL SISTEMA PROPUESTO

Como se mencionó en el Capítulo 2, el sistema se basa en una topología de malla y el monitoreo de la red se lleva a cabo mediante una conexión serial a una computadora.

Componente Alámbrica

La parte alámbrica de nuestra red consta de dos partes principales: la computadora desde la cual se lleva a cabo el monitoreo y la puerta de enlace MIB510CA, que es la que permite retransmitir los datos recibidos mediante un puerto serie RS-232. Cabe recordar que la placa MIB510CA también nos proporciona el enlace necesario para que las aplicaciones desarrolladas en la computadora puedan ser programadas en los motes Mica2 y Mica2DOT.

Adicionalmente es necesario contar con un nodo recolector interconectado a la plataforma MIB510, el cual funcionará como estación base de la red.

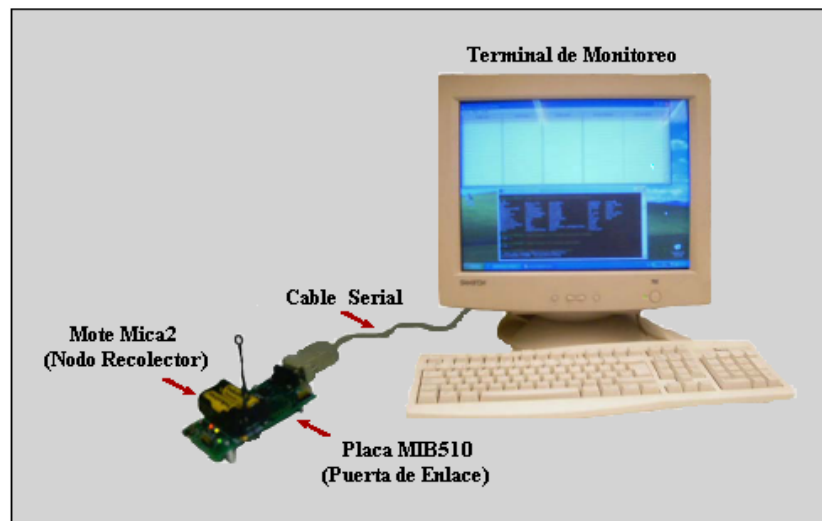


Figura 5.1: Componente Alámbrica de la Red.

Componente Inalámbrica

La parte inalámbrica se conforma de los nodos sensores esparcidos en el campo de prueba. Cada sensor, como hemos visto en capítulos anteriores, se encuentra resguardado en su caja protectora y se conforma de la tableta sensora adherida al mote (Figura 5.2).

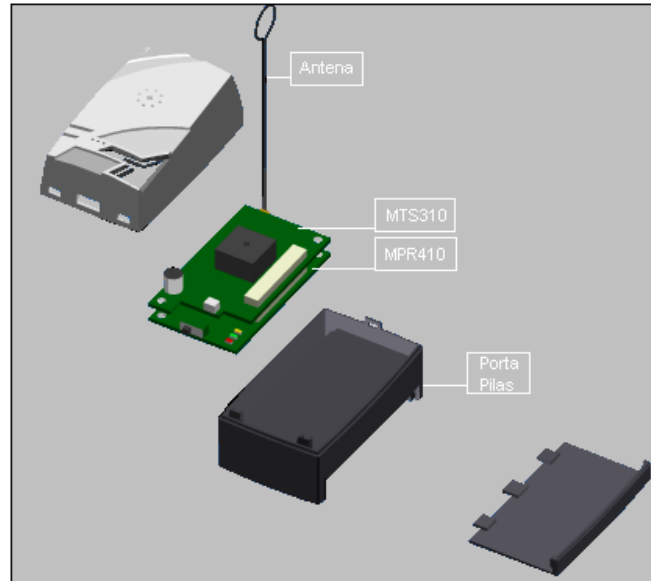


Figura 5.2: Nodo Sensor.

5.1. Colocación de los sensores

La disposición de los sensores se llevó a cabo tomando en cuenta los resultados obtenidos en el Capítulo 4. Considerando la Tabla 4.3 los sensores se colocaron a distancias menores o iguales a 5 [m]. Además se procuró que entre ellos no existieran obstáculos que afectaran la calidad de la señal (según la Tabla 4.4).

La Figura 5.3 muestra la disposición de los sensores dentro del laboratorio de pruebas. El número de nodo representa la *id* con que fueron programados cada uno de los motes. Estos números nos permiten relacionar los datos de sensado desplegados por Mote-View o Surge-View con los nodos localizados físicamente dentro del entorno de monitoreo.

Para la implementación de este proyecto se contaba con cinco tarjetas sensoras: tres tarjetas MTS310 y dos MTS510. Las primeras se colocaron en los nodos: recolector o nodo 0, 2 y 3; y las segundas en los nodos 5 y 6. Mientras que los nodos 4 y 7 sólo desempeñan funciones de redundancia.

Como se observa en la Figura 5.3, el nodo 6 se encuentra en el centro del laboratorio, mientras que el nodo 5 se localiza al lado derecho de la puerta principal. Estos nodos serán los encargados de monitorear los niveles de iluminación dentro del entorno y por ello se colocaron en dos puntos estratégicos: al centro y al fondo de la sala. Aunque los nodos 2 y 3 también sensan el parámetro de iluminancia¹, éstos se encuentran cerca de las ventanas por lo que es probable que sigan percibiendo niveles de luz óptimos mientras que en el fondo de la oficina los valores de iluminancia ya no sean los adecuados para una buena visión.

¹ El nivel de iluminación o iluminancia se define como el flujo luminoso incidente por unidad de superficie.

Los nodos encargados de sensar la temperatura (0, 2 y 3) se ubicaron al centro y a las orillas de la oficina. Los nodos 2 y 3 se ubicaron cerca de las ventanas con el propósito de que percibiesen rápidamente el cambio de temperatura ocasionado por ráfagas de viento o rayos de sol. Por otro lado, el nodo recolector puede detectar un aumento de temperatura ocasionado por una elevada masa corporal dentro del entorno de monitoreo, o bien, un descenso en la temperatura durante las mañanas o temporadas de invierno; periodos en los cuales se percibe un fuerte calor bajo los rayos del sol, mientras que en la sombra se detectan temperaturas bajas.

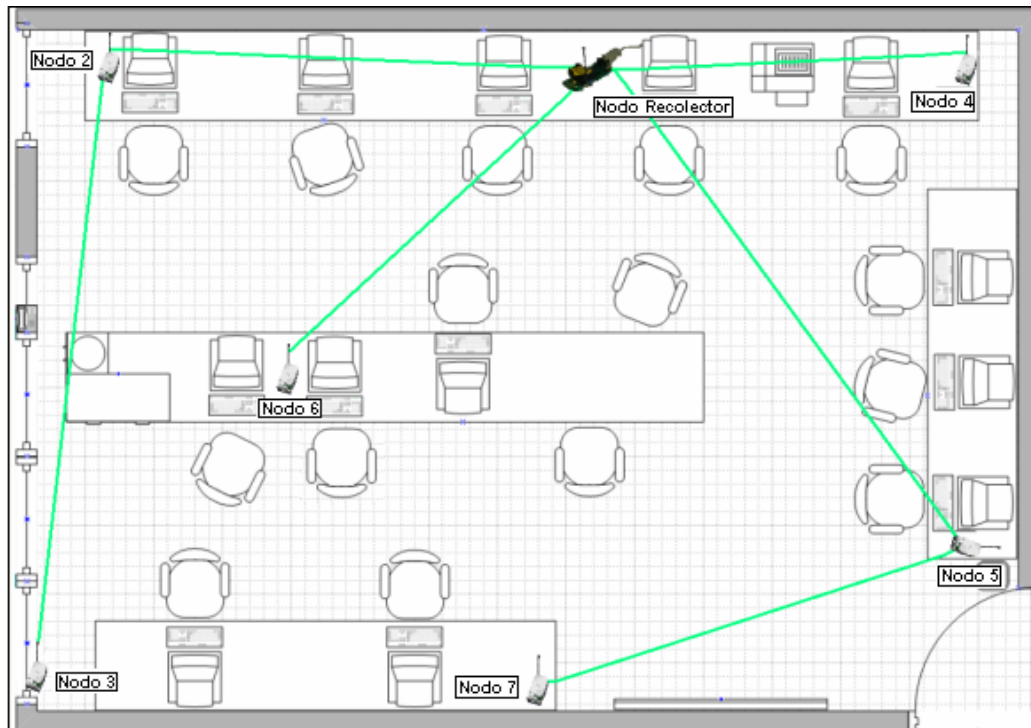


Figura 5.3: Disposición de los Nodos Sensores dentro del Laboratorio de Pruebas.

Las líneas de enlace entre nodos, Figura 5.3, representan la ruta escogida por el protocolo de enrutamiento en el momento de impresión de la imagen, sin embargo no simbolizan el estado de conexión de la red en todo momento.

Tal como se mencionó en el Capítulo 4, el programa *Surge_Reliable* (basado en XMesh), es el encargado de establecer la conexión entre los nodos, siguiendo la trayectoria más óptima de acuerdo a la distancia, calidad de conexión de nodo a nodo, tiempo de respuesta, etcétera.

5.2. Programación y Monitoreo

Como vimos en el capítulo anterior, Mote-View 1.2 cuenta con un gestor de alertas que permite al usuario definir una condición de alarma basándose en cualquier dato sensado desde cualquier nodo de la red. El gestor de alarmas, en conjunto con otras características del software Mote-View, nos permiten llevar un monitoreo eficiente de la red.

Los motes deben programarse con alguna aplicación compatible con Mote-View. Algunas de estas aplicaciones son: XMesh, XSensor, Surge_Reliable y Surge_Reliable_Dot; siendo estas dos últimas las aplicaciones utilizadas en este proyecto.

Los motes Mica2 y Mica2Dot son programados con ayuda de la plataforma MIB510 desde una ventana de cygwin. La configuración del archivo *MakeXbowlocal* (localizado en el directorio *opt/tinyos-1.x /contrib/xbow/apps/*), se realiza previamente a la descarga del programa en los motes ya que en él se establecen:

- La plataforma desde la cual se van a programar los sensores (MIB510 ó MIB600).
- El puerto común utilizado por la plataforma (en nuestro caso COM1).
- ID del grupo local de Red (en nuestro caso 0x7D en hexadecimal, 125 en decimal).
- El Canal de Comunicación (frecuencia de transmisión /recepción).
- La Potencia de Transmisión de la señal de RF (en este proyecto: 0x0b, igual a -3 [dBm]).

Antes de instalar la aplicación ésta debe construirse tecleando las siguientes instrucciones: *make mica2* o *make mica2dot*, en el directorio correspondiente según el mote en el que se desee instalar (Surge_Reliable ó Surge_Reliable_Dot). Posteriormente se adhiere el mote a la placa MIB510 (en el conector de 51 pines para el mote mica2, o en el conector de 19 pines para el mote mica2dot), la cual es conectada previamente a la computadora mediante el cable serial.

La aplicación se instala desde los directorios *opt/tinyos-1.x/contrib/xbow/apps/Surge_Reliable* u *opt/tinyos-1.x/contrib/xbow/apps/Surge_Reliable_Dot*, dependiendo del mote que desee programarse. Para ello pueden teclearse las siguientes instrucciones: *make mica2 install.<nodeid>* ó *make mica2 reinstall.<nodeid>*, para los motes mica2 ; ó *make mica2dot install.<nodeid>* ó *make mica2dot reinstall.<nodeid>*, para los mote mica2dot. Las instrucciones *install* y *reinstall* difieren en que la segunda instrucción ya no compila la aplicación antes de instalarla, de manera que se realiza con mayor rapidez; sin embargo, si el programa tuviese algún error de sintaxis, esta instrucción no detectaría ningún fallo.

La etiqueta *<nodeid>* corresponde al número identificador que se le asigna a cada nodo para ubicarlo dentro de la red. Cabe mencionar que el nodo recolector (que permanece en la plataforma MIB510) debe instalarse con un *<nodeid>* de 0. Por otro lado, esta etiqueta nunca deberá ser 0 para un mote Mica2Dot, y deberá ser único y diferente para cada nodo de la red, sean Mica2 o Mica2Dot.

Las pruebas de fiabilidad, calidad de enlace, cobertura, etcétera; se efectuaron con ayuda de la aplicación Surge-View; mientras que el monitoreo de la red se llevó acabo con la

aplicación Mote-View debido a que proporciona mayor información sobre la red y mejores herramientas para su manejo.

La aplicación Surge-View se ejecuta con ayuda del programa SerialForwarder, el cual es utilizado para leer los paquetes de datos desde el puerto serial de la computadora y remitirlos a una conexión de puerto utilizada por el servidor. Dicho programa permite que la aplicación en cuestión, además de otros programas, pueda comunicarse con la red de sensores vía una puerta de enlace (MIB510). Este programa puede ejecutarse de dos maneras: 1) en una ventana de comandos MS-DOS, con la siguiente instrucción: *SerialForwarder -comm serial@COM1:mica2*; o bien 2) ejecutando el archivo SerialForwarder.exe, desde la carpeta *C:\Archivos de programa\Surge-View*.


Finalmente, la aplicación Surge-View se activa desde una ventana de comandos de MS-DOS en el directorio *c:\Archivos de Programa\Surge-View* con la siguiente instrucción: *surge 125*. El número 125 representa la ID del grupo local de red (previamente definido en el archivo MakeXbowlocal). Cabe mencionar que aunque dos motes se encuentren en el mismo canal no podrán comunicarse si no están dentro del mismo grupo ID.

Monitoreo con Mote-View

Conexión a la Red de Sensores

Una vez instalada la aplicación Surge_Reliable:

1. En la barra del menú principal se selecciona la opción: *File > Connect to MIB510/MIB600*. Como estamos utilizando la plataforma MIB510, la tasa de transmisión es de 57600 [bps] y en nuestro caso el puerto serial es COM1.
2. En el recuadro de aplicaciones XMesh se selecciona Surge_Reliable (aplicación con la cual han sido programados los motes).
3. Otras opciones disponibles son:
 - **Despliegue de datos en bytes (-r)**. Paquetes de bytes ingresando en la PC.
 - **Despliegue de datos analizados sintácticamente (-p)**. Paquetes de datos provenientes del ADC.
 - **Despliegue de los datos convertidos (-c)**. Los datos que ingresan a la computadora son convertidos a unidades ingenieriles.
 - **Inicialización en la Base de Datos (-l)**. Almacenará los valores provenientes del ADC a la base de datos *PostgreSQL*.
4. Para que los datos se visualicen en tiempo real es necesario seleccionar el recuadro *Live*, localizado en el menú principal. Y tomar en cuenta cualquier dato desplegado en la ventana *Server Messages*, situada en la parte inferior de la ventana de Mote-View.

5. Con el fin de almacenar los datos extraídos de la red, es necesario conectarla a una base de datos localizada en un servidor local o remoto:
 - a) En la barra de menú principal: click izquierdo del mouse sobre el icono de conexión . La mínima información requerida es: **Server**, **Port**, **Username** y **Password**. Mientras el servidor sea local, es decir, mientras la información sea recabada mediante la plataforma MIB510 en la misma máquina en la que se monitorea, estos campos contendrán valores determinados por el fabricante y no deberán ser modificados, como es el caso de nuestra aplicación.
 - b) Click izquierdo sobre el botón Connect, para conectar Mote-View con el servidor dado.
 - c) Una vez seleccionadas la base de datos y el nombre de la tabla, se inicia la conexión y se abre una ventana MS-DOS, en donde se observan los datos provenientes de la red. La base de datos seleccionada es **task**, mientras que el nombre de la tabla es **surge_results**, ya que los resultados provienen de Surge_Reliable (programa previamente instalado en los notes).

Establecimiento de Alarmas

Para los fines de este proyecto se utilizó el gestor de alarmas de MOTE-VIEW. Un alerta se compone de varias piezas:

- **Identificador de Alerta (ID)**. Un número secuencial que ayuda a identificar la alerta. Inicia en uno y aumenta de manera secuencial según se vayan agregando nuevas alertas.
- **Nombre del Nodo**. El nodo cuyas condiciones puede activar una alerta.

Los nodos que activarán las alertas en nuestro sistema de monitoreo son: Nodo 0, Nodo 2, Nodo 3, Nodo 5 y Nodo 6.

En el caso del monitoreo de temperatura se requieren dos alertas, una que activaría el aire acondicionado y otra que activaría el calefactor. A su vez, cada alerta requiere de dos estados, cuando se está *por debajo* o *por encima* de la temperatura deseada. Así pues, como el Nodo 0 sólo detectará temperatura, éste deberá repetirse cuatro veces.

Los Nodos 2 y 3 se repetirán cinco veces debido a que ellos establecerán condiciones de alerta tanto para temperatura como para iluminancia; mientras que los nodos 5 y 6 sólo se escribirán una vez, ya que para la iluminancia sólo se requiere una condición.
- **Nombre del Sensor**. El sensor cuyas lecturas desencadenan una alerta.

En el Nodo 0, el sensor que desencadenará las alertas es la temperatura; mientras que en los Nodos 2 y 3, los sensores serán cuatro de temperatura y uno de luz, ya que

cada uno requiere cuatro alertas para el control de temperatura y una para el control de luminancia.

Los Nodos 5 y 6, como encargados de monitorear los niveles de iluminancia dentro del entorno, activaran alarmas mediante el sensor de luz.

➤ **Condición de alerta.** Operación de comparación para decidir que tipo de alerta será desencadenada:

- Si existe una igualdad
- Si se ha excedido un valor
- Si se está por debajo de un valor determinado
- Si la condición es igual o mayor a la establecida
- Si la condición es igual o menor a la establecida

Para el Nodo 0 las condiciones son:

- $<$ (menor que 20.56 [°C]) y $>$ (mayor que 20.56 [°C])
- $>$ (mayor que 23.30 [°C]) y $<$ (menor que 23.30 [°C])

Para los Nodos 2 y 3 se incluyen las cuatro condiciones anteriores y se anexa la siguiente:

- $<$ (menor que 200 [lux])

Ésta última condición será la misma para los Nodos 5 y 6.

➤ **Umbral de alerta.** El valor contra el cual se compara la condición de alerta.

Para los valores de temperatura de los Nodos 0, 2 y 3:

- **20.56** (mayor y menor que 20.56 [°C])
- **23.30** (mayor y menor que 23.30 [°C])

El valor para el nivel de iluminación de los Nodos 2, 3, 5 y 6:

- **200** (menor que 200 [lux])

➤ **Unidad.** La unidad para el umbral de alerta.

La unidad utilizada para la temperatura es grados Celsius, mientras que el sensor de luz, aunque no presenta ninguna opción de unidad, se deduce que es el *lux* (unidad para el nivel de iluminación o iluminancia), puesto que los valores son coherentes de acuerdo a lo establecido como exigencias visuales en el Real Decreto 486/1997, citado por la Unión General de Trabajadores (UGT) en su fascículo 22 de Salud Laboral (citado anteriormente).

- **Acción que desencadena una alerta.** Es la operación que se ejecuta en respuesta a una activación de alerta. Existen dos respuestas posibles ante una activación de alerta:
 - Desplegar una ventana de diálogo con el mensaje de alerta.
 - Enviar un e-mail a una dirección previamente definida.

Se escogió el despliegue de una ventana de diálogo como tipo de acción ha desencadenarse.

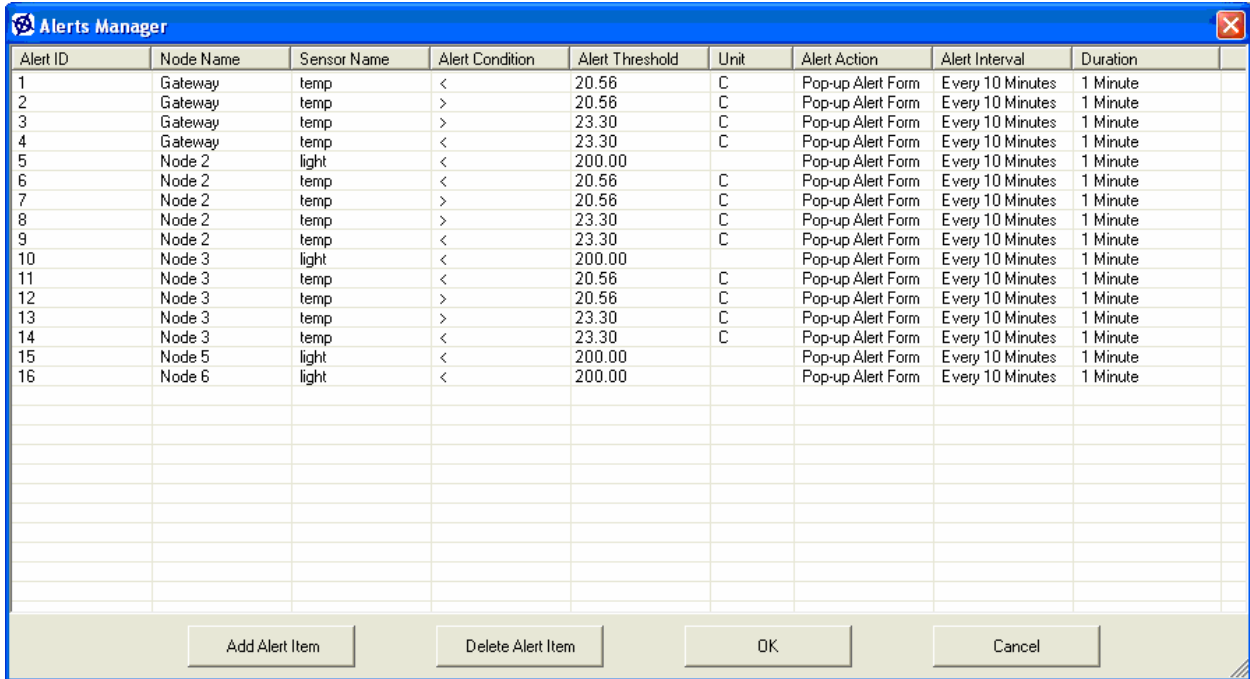
- **Intervalo de Alerta.** Especifica el intervalo durante el cual debe ser ejecutada la acción.

El intervalo de alerta se fijo en diez minutos, de manera que la alerta se estará ejecutando cada diez minutos hasta que la condición establecida vuelva a su estado óptimo.

- **Tiempo de duración.** Es el período durante el cual debe cumplirse la condición de alerta antes de que ésta sea activada.

Se determinó un tiempo de duración de un minuto; esto es, que debe rebasarse el umbral establecido durante el intervalo de un minuto antes de que la alarma se active.

Finalmente la tabla de alertas queda como se muestra en la Figura 5.4.



Alert ID	Node Name	Sensor Name	Alert Condition	Alert Threshold	Unit	Alert Action	Alert Interval	Duration
1	Gateway	temp	<	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
2	Gateway	temp	>	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
3	Gateway	temp	>	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
4	Gateway	temp	<	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
5	Node 2	light	<	200.00		Pop-up Alert Form	Every 10 Minutes	1 Minute
6	Node 2	temp	<	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
7	Node 2	temp	>	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
8	Node 2	temp	>	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
9	Node 2	temp	<	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
10	Node 3	light	<	200.00		Pop-up Alert Form	Every 10 Minutes	1 Minute
11	Node 3	temp	<	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
12	Node 3	temp	>	20.56	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
13	Node 3	temp	>	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
14	Node 3	temp	<	23.30	C	Pop-up Alert Form	Every 10 Minutes	1 Minute
15	Node 5	light	<	200.00		Pop-up Alert Form	Every 10 Minutes	1 Minute
16	Node 6	light	<	200.00		Pop-up Alert Form	Every 10 Minutes	1 Minute

Figura 5.4: Alarmas Establecidas.

De manera que si la temperatura al momento de encender el sistema es de 25 [°C], y se mantiene mayor a 23.30 [°C] durante un minuto, esto activará una alerta, la cual mediante un sistema mecánico activaría el aire acondicionado. Dicha alerta se repetirá cada diez minutos mientras la temperatura siga siendo mayor a 23.30 [°C]. Cuando el valor de la temperatura sea de 23.29 [°C] y se mantenga menor a 23.30 [°C] por un periodo de un minuto, se activará una segunda alerta, con la que se controlaría el apagado del aire acondicionado.

Es importante notar que el tiempo de duración es un factor esencial al momento de implementar un sistema mecánico, ya que éste nos permitiría aumentar o disminuir la frecuencia entre el encendido y el apagado del sistema, con lo que se lograría reducir el desgaste físico del mismo

Así mismo, si la temperatura al momento de encender el sistema es de 17 [°C] y se mantiene menor a 20.56 [°C] durante un minuto, se activará una alerta (con la que se activaría un sistema mecánico que encendería un calefactor). Si en diez minutos la condición de temperatura no ha cambiado, es decir sigue siendo menor a 20.56 [°C], el sistema seguirá activando la alerta. Al momento de alcanzar una temperatura igual a 20.57 [°C] se inicia un conteo, el cual seguirá mientras la temperatura se mantenga por encima del umbral (20.56 [°C]). Si el tiempo de conteo transcurrido es igual a un minuto, entonces otra alerta será activada (ésta hace referencia al apagado del calefactor).

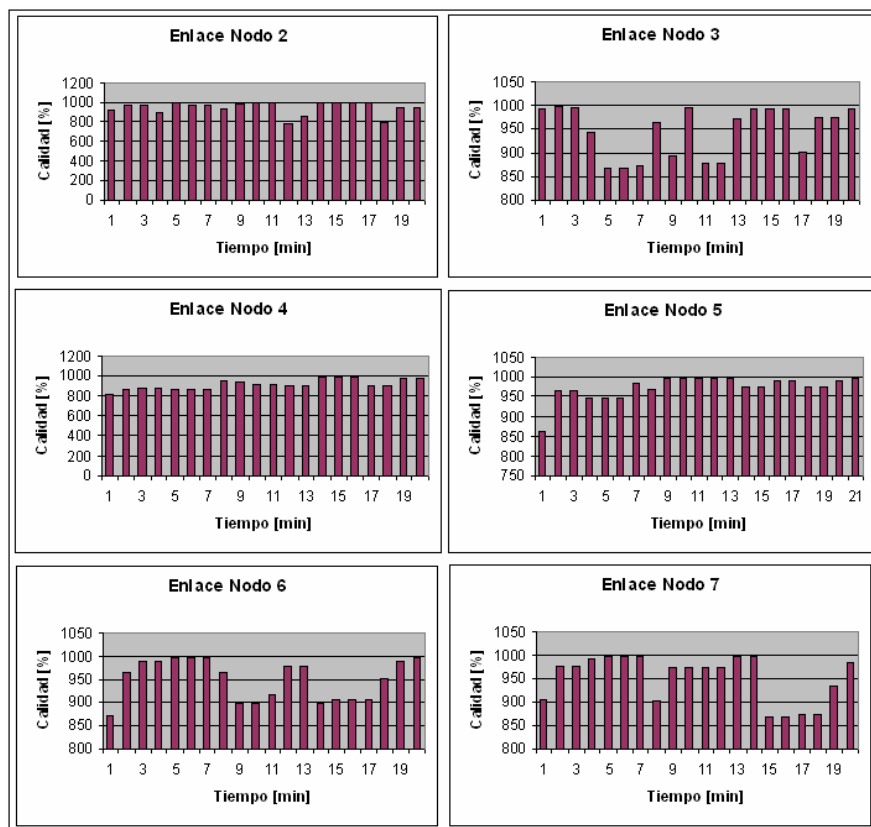
5.3. Pruebas sobre el sistema

Hasta este punto, se han programado los motes, indicado las alarmas correspondientes para cada uno de ellos y establecido su posición en el espacio de prueba. El siguiente paso consiste en determinar la efectividad de la red dispuesta mediante diversas pruebas de campo.

En este análisis experimental se consideraron dos escenarios distintos. En el primer caso el laboratorio de pruebas se encontraba en una situación óptima para el uso del sistema, esto es, un número moderado de usuarios (aproximadamente 6 usuarios). En el segundo escenario, el medio era poco recomendable, esto es, un alto número de usuarios (15 usuarios aproximadamente).

Escenario 1

Las Gráficas 5.1 muestran el porcentaje de calidad de los enlaces en la red, mientras que el comportamiento de conectividad entre nodos se observa en la Figura 5.5.



Gráficas 5.1: Calidad de Enlaces en la Red; Escenario 1.

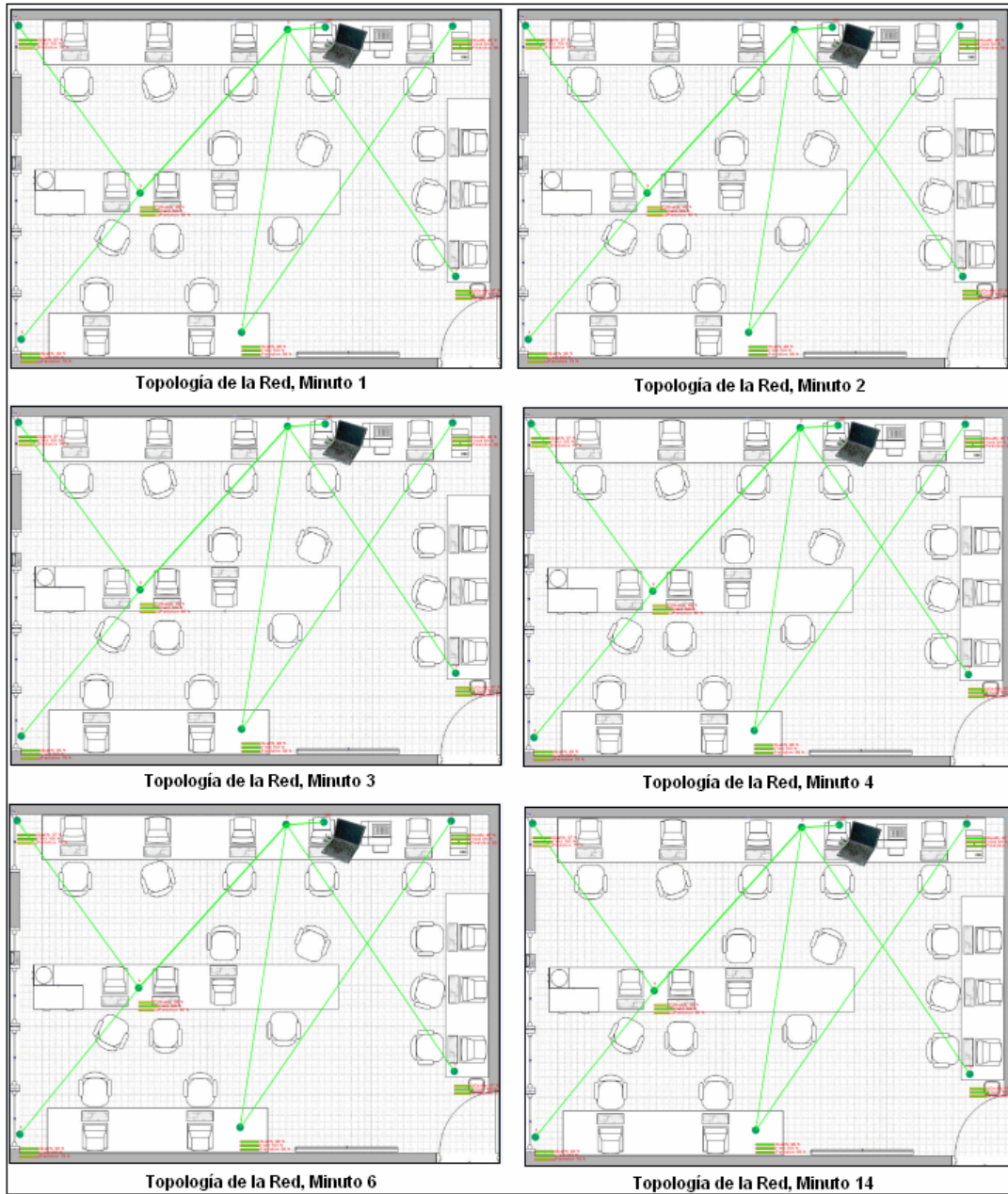
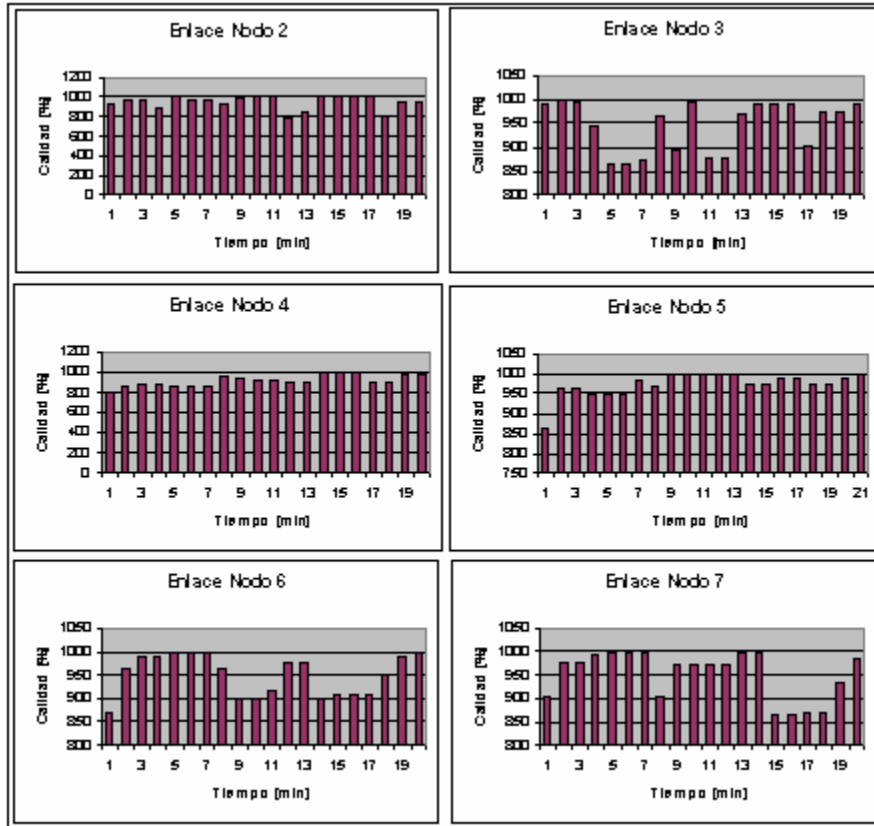


Figura 5.5: Conectividad entre Nodos, Escenario 1.

Escenario 2

En las Gráficas 5.2 se observan la calidad de los enlaces para este escenario, mientras que la Figura 5.6 muestra el comportamiento de la conectividad entre nodos.



Gráficas 5.2: Calidad de Enlaces en la Red; Escenario 2.



Figura 5.6: Conectividad entre Nodos, Escenario 2.

De las Figuras 5.5 y 5.6 podemos percibir claramente que la conectividad entre nodos varía constantemente cuando existe un gran número de usuarios dentro del laboratorio de pruebas, mientras que, para un número reducido la topología se mantiene constante. En la segunda figura, la topología de la red va cambiando de acuerdo a la ruta que cada nodo considere adecuada, pues como sabemos cada nodo posee funcionalidades de enrutamiento y host, incluyendo descubrimiento de rutas (Multi-Hop) y entrega de mensajes. Por consiguiente, la topología de la red es dinámica. Este efecto se produce

por interferencias cambiantes (usuarios moviéndose en el entorno de prueba), de manera que tenemos nodos conectándose y otros siendo desconectados, ya sea entrando al rango de otros nodos o alejándose.

La dispersión utilizada en nuestra red inalámbrica (radiación isotrópica), hace que la señal transmitida rebote en techos, paredes y en este caso personas, introduciendo un efecto de interferencia en el receptor, lo cual no sólo limita la calidad del enlace, sino que afecta la velocidad de transmisión ya que la trayectoria reflejada llega con un retraso al receptor.

De cualquier forma, como podemos observar en las Gráficas 5.1 y 5.2, la calidad de la señal se mantiene en un nivel adecuado a lo largo de estas pruebas (no menor al 80%), con lo cual concluimos que para cada enlace la conexión se conservará hasta el punto en que el vínculo sea considerado complejo o imposible y el nodo opte por encaminar la señal en otra dirección, siempre buscando conservar un porcentaje elevado en la calidad de la señal.

En lo que a monitoreo de la red se refiere; las Figuras 5.6 y 5.7 presentan las gráficas correspondientes a las magnitudes de luz y temperatura medidas en los nodos que componen la red. En tanto que las Figuras 5.8 y 5.9 exhiben una topología de la red implementada en la que se visualizan los valores de temperatura y luz detectados por cada nodo.

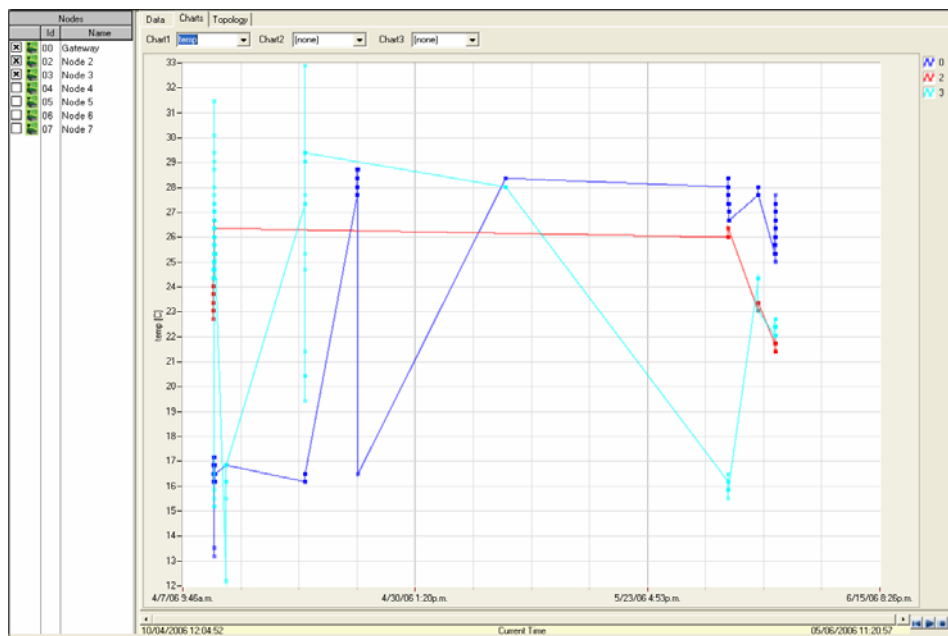


Figura 5.7: Gráficas de Temperatura para los Nodos 0, 2 y 3.

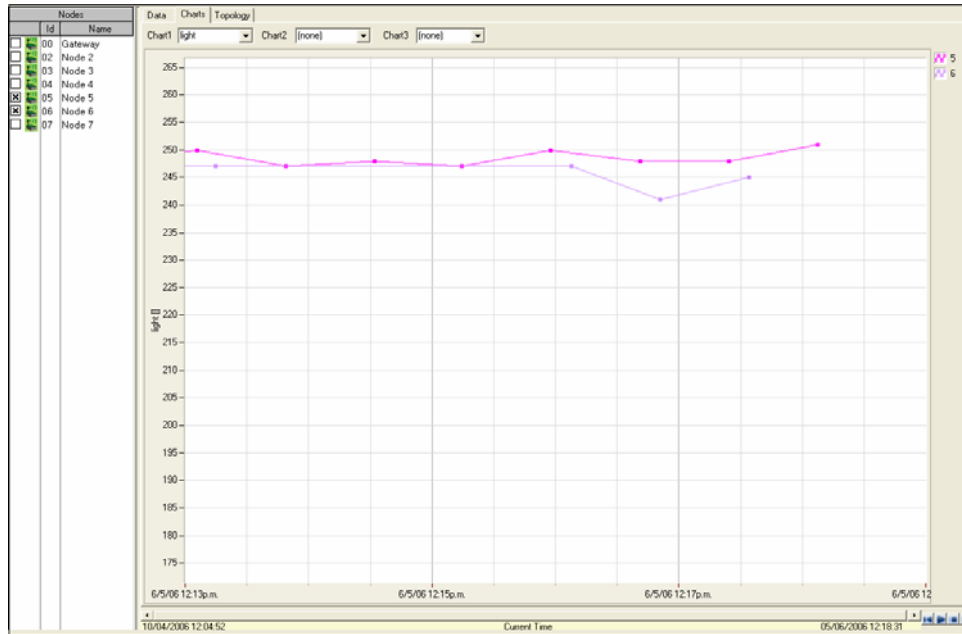


Figura 5.8: Gráficas de Luz para los Nodos 5 y 6.

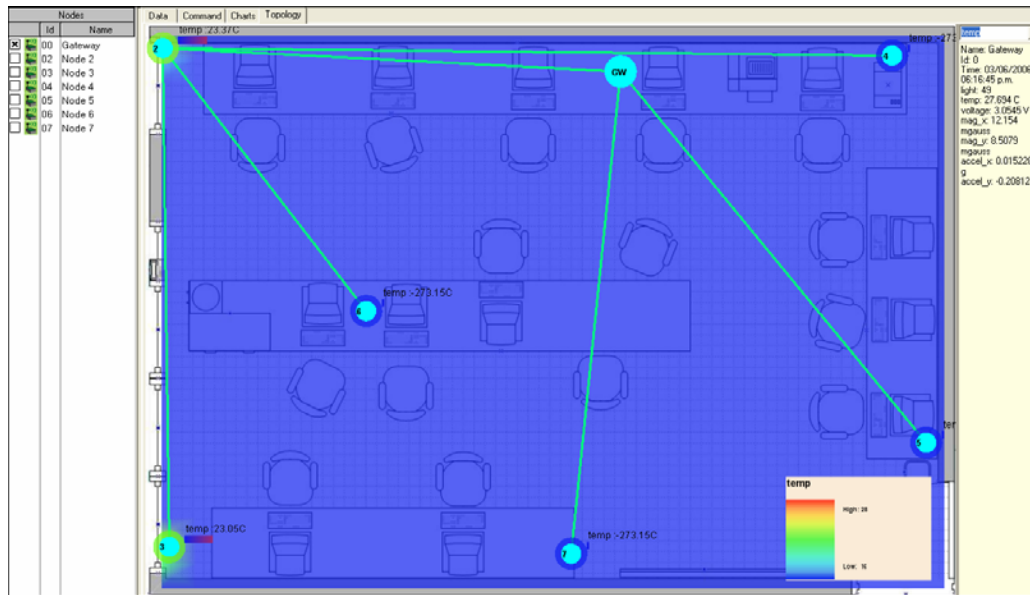


Figura 5.9: Topología de la Red; Niveles de Temperatura.

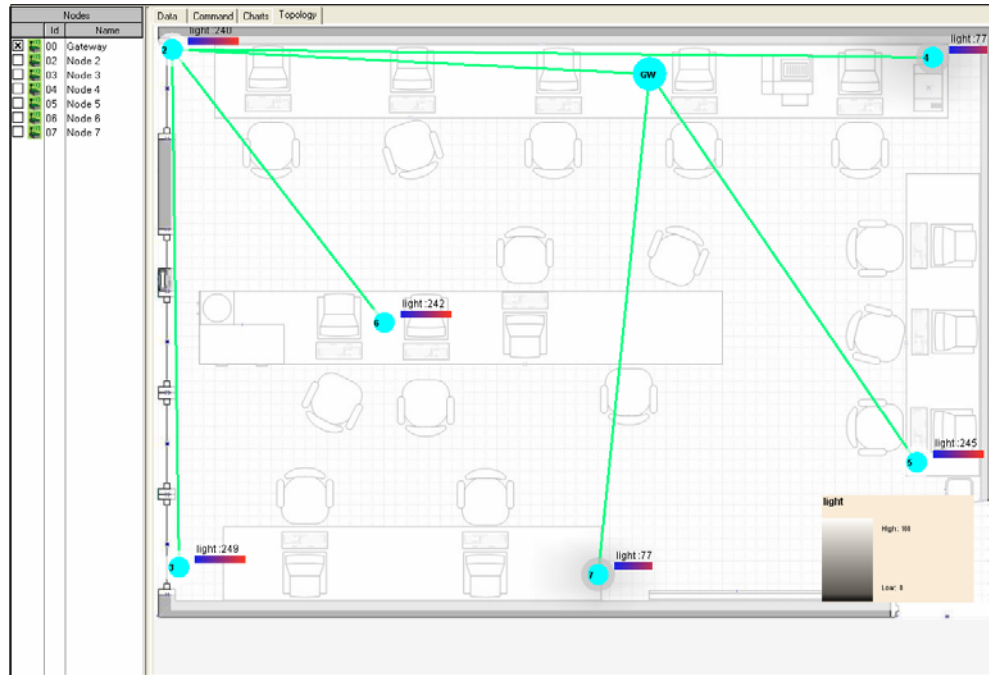


Figura 5.10: Topología de la Red; Niveles de Luz.

Las figuras anteriores son una muestra de la practicidad con que una red dispuesta puede ser monitoreada. En otro tipo de aplicaciones éstas gráficas podrían ser de gran utilidad para establecer un sin fin de patrones, desde determinación de horarios de acuerdo a los niveles de luz registrados hasta el diagnóstico de climas en una determinada zona, o bien de temperaturas en estructuras y equipos industriales.

5.4. Aplicaciones del Sistema Implementado

El sistema estudiado en este proyecto tiene como aplicación principal el monitoreo de una oficina, sin embargo las validaciones hechas comprueban la funcionalidad de cada uno de los componentes de la red, abriendo así la posibilidad de utilizar dichos elementos en la implementación de cualquier otro servicio, o bien ampliar la aplicación existente y con ello su eficacia y cometidos. Así por ejemplo, como se indicó en el Capítulo 2, pueden incorporarse al sistema de alarmas de la red los servicios de aire acondicionado y calefacción.

Las aplicaciones de este sistema, como de cualquier red de sensores inalámbricos, es tan extensa como la imaginación del usuario ya que monitorear y manipular elementos físicos presenta infinidad de posibilidades en casi todos los campos de la ciencia. Por mencionar algunas de las aplicaciones posibles y limitándonos a las magnitudes utilizadas en éste proyecto tenemos: 1) *Monitorizar Temperatura en:* pacientes, viñedos, bodegas de alimentos ó medicamentos, hábitats artificiales, etcétera. 2) *Monitorizar Luz en:* alumbrado en calles, bodegas, edificios corporativos, microsistemas, etcétera.

El objetivo del monitoreo periódico en cualquier sistema consiste en identificar el deterioro incipiente de las condiciones consideradas como óptimas, ya sea de una estructura, entorno ambiental, hábitat, etcétera; de manera que se puedan tomar decisiones oportunas y reducir la probabilidad de fallas catastróficas.

En este proyecto la identificación de los parámetros sensados no se efectuó para evitar fallas catastróficas sino para ofrecer diversas ventajas a sus usuarios. En primera instancia proporciona las utilidades de cualquier sistema de monitoreo de oficinas, esto es, anticipar una necesidad y actuar en favor nuestro satisfaciendo el requerimiento en cuestión, disminuyendo así la fuerza de trabajo y el tiempo empleados en ella.

Por otro lado, entre las características que lo distinguen de los sistemas convencionales se encuentran: facilidades de desarrollo, instalación y mantenimiento, utilización de hardware de bajo costo y finalmente, requerimientos de energía e infraestructura mínimos.

En términos más específicos, el monitoreo de una oficina permite que el desempeño de un individuo en su trabajo sea lo mejor posible y se convierta en beneficio para su organización, ya que del grado de confort que tenga una persona en su lugar de trabajo dependerá su estabilidad física, mental y social.

Capítulo 6

CONCLUSIONES

6.1. Conclusiones

De acuerdo a los resultados obtenidos, tanto en el Capítulo 4 en donde se validaron cada una de las componentes del sistema, como en el quinto capítulo, en el cual se realizaron pruebas sobre el sistema implementado; podemos hacer las siguientes observaciones:

- Es factible desarrollar un sistema de monitoreo de oficina eficaz, mediante la utilización de una red híbrida del tipo Infraestructura y Ad-Hoc con sensores inalámbricos.
- Este sistema cumple con los requerimientos de cualquier sistema de monitoreo:
 - Rápida detección y respuesta frente a un problema determinado,
 - Señales de alta calidad (mayores al 80% según las gráficas obtenidas),
 - Nodos redundantes con alta conectividad y poca vulnerabilidad (hecho que nos permite contar con una red fiable), y finalmente
 - Una interfaz de usuario simple, práctica y completa.

De manera que los objetivos planteados al inicio de esta investigación se cumplieron de manera satisfactoria, al recrear las condiciones bajo las cuales funciona el sistema de monitoreo implementado.

6.2. Contribuciones

La contribución principal de este documento es la descripción de una tecnología emergente en el campo de la investigación.

Las Redes de Sensores Inalámbricas o WSN son una nueva y prometedora tecnología basada en recientes avances en los campos de las telecomunicaciones, microelectrónica y software para sistemas empotrados. Estas tecnologías permiten que varios nodos puedan comunicarse entre sí, sin la existencia de un punto central de coordinación, ya que cada nodo tiene funciones de ruteo. La comunicación puede realizarse en cualquier dirección, lo que facilita la existencia de rutas redundantes de comunicación.

Las aplicaciones de estas redes pueden considerarse casi ilimitadas, específicamente en sistemas de control, monitoreo y comunicación; ya que adicionalmente a las características de los sistemas convencionales, éstas proporciona muchas otras ventajas, entre ellas podemos mencionar las siguientes:

- Se pueden desplegar de forma flexible en entornos que no disponen de infraestructura fija,
- Los componentes de la red requieren de poca energía para su función,
- El hardware utilizado es de bajo costo,
- Son fáciles de desarrollar, instalar y mantener,

- Reducen costos de cableado y tiempo de instalación, y
- Permiten movilidad, portabilidad y regeneración de la red

Este documento proporciona además, una guía básica a estudiantes de licenciatura y/o maestría que pretendan estudiar los conceptos de una WSN, utilizando hardware de la compañía Cross-Bow y el sistema operativo TinyOS.

Día con día existe un mayor número de redes de sensores. Dentro de las redes Ad-Hoc, éste tipo de redes son las que tienen mayor número de usos civiles; esto desde finales de los años 90 y principios del siglo 21, cuando los sensores empezaron a evolucionar, decreciendo en tamaño e incrementando su autonomía.

El futuro ya ha empezado a ser escrito por otra compañía llamada Dust Inc, compuesta por miembros del proyecto Smart Dust ubicado en Berkeley, la cual ha creado nodos de un tamaño inferior al de un balón y que debido a su minúsculo tamaño implicarán múltiples nuevas aplicaciones.

A continuación se enuncian algunas de las aplicaciones recientes en diversas partes del mundo, las cuales pueden ser la pauta para trabajos futuros y que ratifican la trascendencia de estas redes en el campo tecnológico, social y económico.

- En Suiza, científicos del Instituto de Investigación Agroscópico de Changins-Wädenswil, examinan la funcionalidad de la red de sensores sin cable en el viñedo ubicado en esta institución. El sensor planea ser implementado en la India para ayudar a los campesinos a determinar con más precisión cuánta agua necesitan sus cultivos. Se estima que se podría ahorrar un 40 por ciento del agua utilizada para la irrigación de los sembradíos si se dispusiera de información más fiable sobre la demanda acuífera. El detector ha sido desarrollado por el Polo de Investigación Nacional (PIN), «Sistemas móviles de información y comunicaciones» en colaboración con el Instituto Indio de Ciencias en Bangalore.
- La universidad de Viña del Mar (UVM), la municipalidad de Catemu y CORFO, planean incluir en el proyecto tecnológico “Red Wireless Rural Catemu” (RWR-C), la primera red rural de comunicación inalámbrica en Chile; una red interconectada de sensores agroclimáticos. Esta red pretende brindar una aplicación web para analizar los campos y la gestión agrícola; basándose en la premisa de que la información digital mantenida de manera continua pasa a ser una pieza clave en los procesos de producción, procesamiento y comercialización de los productos. La información impacta en los sistemas de gestión y en la capacidad de reacción de las empresas, lo cual repercute en los costos, en la calidad de la producción, en el manejo de los riesgos y en la sustentabilidad ambiental de los procesos.
- Los laboratorios de SUN MICROSYSTEMS, en España, desarrollan redes de sensores inalámbricos usadas conjuntamente con etiquetas RFID y mecanismos de activación. Entre las aplicaciones que ya están en desarrollo destacan la detección de defectos de fabricación mediante la búsqueda de vibraciones extrañas en

equipamiento industrial, la monitorización de los movimientos de los pacientes en un hospital, seguimiento de vertidos de fertilizantes en lagos y la determinación del origen de un disparo.

- Un equipo investigador de la Universidad del estado norteamericano de Iowa está desarrollando una red de sensores inalámbricos que podría dirigir el sistema de transmisión de electricidad en Estados Unidos, según un comunicado de prensa emitido por la universidad en junio del 2006. Los investigadores han avanzado en el desarrollo de un sistema prototípico, dice la nota. Integrados en cajas negras de varias pulgadas de ancho, los sensores captarían actividades sospechosas registradas en postes de electricidad, y resultarían especialmente útiles a la hora de localizar con rapidez cualquier avería. Asimismo, permitirán a las compañías de electricidad una reacción a tiempo para prevenir que los trastornos en el suministro de energía puedan desembocar en apagones, y el sistema de supervisión también ayudaría a las compañías a localizar rápidamente problemas cuando las condiciones meteorológicas adversas dañen las líneas de electricidad.

Con las redes de sensores, "las compañías de electricidad poseerían habilidades adicionales de observación de sus sistemas, y asistirían a las tareas de recuperación de desastres", explicó el principal investigador Arun Somani. El equipo investigador ha comenzado a hablar con las compañías de electricidad sobre la posibilidad de poner a prueba el sistema sobre la red de suministro de electricidad, indicó Somani. Estados Unidos cuenta con muchas líneas de transmisión, subestaciones y generadores que podrían utilizar esos sensores de supervisión.

En las aplicaciones antes expuestas, así como en las ventajas enunciadas de una WSN, radica la importancia de caracterizar el funcionamiento de este tipo de redes, tal y como se ha hecho en el presente trabajo.

Otra motivación ha sido la escasez de documentos y pruebas con sensores reales en nuestro país; hecho que da aún mayor relevancia a las pruebas efectuadas en este proyecto.

6.3. Retos

Debido a la naturaleza de la comunicación inalámbrica, los nodos Ad-Hoc tienen que lidiar con diversos problemas tales como:

Ancho de banda limitado. El ancho de banda disponible en una interfaz inalámbrica es inferior al de una cableada, y además se infrutiliza debido a la atenuación e interferencias de las señales electromagnéticas. A pesar de esto, los usuarios pretenderán utilizar aplicaciones similares a las que ya usan en entornos tradicionales.

Consumo de energía. Un elemento importante para asegurar la fiabilidad de los nodos en las redes inalámbricas es la fuente de poder de los dispositivos que la forman. La naturaleza inalámbrica implica que estos dispositivos deben tener ya sea un bajo consumo o un modo eficiente de conservación de manera que puedan permanecer con su

propia batería por tiempos prolongados. A pesar de los recientes avances en el área de la capacidad de las baterías, el consumo de energía, es todavía un factor de diseño crítico¹.

Calidad del enlace. La calidad de un enlace puede disminuir por efecto de: ruido, interferencias y fluctuaciones de la potencia de la señal debido a variaciones en el medio de transmisión, entre otras cosas.

Protocolos de Enrutamiento. Los diseñadores de protocolos y/o aplicaciones para redes de sensores inalámbricas deben tener presentes las características antes mencionadas, con el fin de mitigar los efectos perjudiciales que pudieran tener. Los protocolos que se implanten deben reaccionar ante cambios topológicos para crear nuevas rutas. No deben ocupar demasiado ancho de banda con mensajes de control, ya que esto disminuye el ancho de banda disponible para tráfico útil, es decir, del nivel de aplicación. El tiempo de procesamiento debe reducirse al máximo, para evitar agotar las baterías de los nodos.

Por ahora, y principalmente dando respuesta a las necesidades crecientes de comunicación es que estos retos están siendo tratados, y se han podido ir reduciendo poco a poco.

6.4. Vías de Aplicación Futuras

Como trabajo futuro inmediato se podría anexar un dispositivo mecánico al sistema de alarmas de monitoreo desarrollado en este trabajo, sin embargo la validación de los elementos utilizados en este proyecto (kit profesional MOTE-KIT5040), brinda un sin fin de posibilidades en la investigación de estos nuevos dispositivos tecnológicos:

- Hay diferentes cuestiones aún por explorar en lo referente a la interconexión entre redes Ad-Hoc y redes fijas,
- Estudiar y cuantificar el efecto que tiene los diversos protocolos de enrutamiento sobre una red de sensores real ,
- Caracterizar parámetros tales como el ancho de banda, latencia, consumo de batería y capacidad de reacción respecto a cambios en la topología provocados por la propia red sensora, etcétera.,
- Estudiar la viabilidad de implementar diversos sistemas de redes de sensores, por ejemplo: sistemas de localización, monitoreo de dispositivos móviles, seguridad, etcétera.,
- Investigación y análisis de cobertura y conectividad en redes de sensores inalámbricas móviles, etcétera.

¹ Shingo Tsuda, “*Battery Technologies Based on the Requirements for Applications and Future Movement*”, 10th Int. Meeting on Lithium Batteries, June 2, 2000, Como (Italy).

REFERENCIAS

- [1] "Getting Started Guide" de Cross-Bow
- [2] "MPR-MIB Series User Manual" de Cross-Bow
- [3] "MOTE-VIEW 1.2 Users Manual" de Cross-Bow
<http://www.xbow.com>
- [2] TinyOS Tutorial
<http://www.tinyos.net/tinyos-1.x/doc/tutorial/>
- [3] Andrew S. Tanenbaum "Redes de computadoras" Cuarta edición.
- [4] Francisco Javier Ros Muñoz "Evaluación de Propuestas de Interconexión a Internet para Redes Móviles Ad Hoc Híbridas", Diciembre 2004.
- [5] Hasan Ozer, "Wireless Sensor Networks for Crack Displacement Measurement" , Julio 2005.
- [6] "Sensor Network – Basic Function & Validation"
<http://www.itu.dk/people/hammad/sensorNetwork/report.doc>
- [7] "Tecnologías y actividades de estandarización para la interconexión de Home Networks"
http://www.fundacionauna.org/areas/26_estudios/pdf/3.pdf
- [8] "Tecnologías para redes LAN inalámbricas y Windows XP"
<http://www.microsoft.com/latam/windowsxp/pro/biblioteca/planning/wirelesslan/intro.asp>

APÉNDICE

Apéndice A: Funcionalidad Multi-Hop

La red Mesh de Cross-Bow se forma en base a un componente de software escrito en TinyOS. TinyOs (introducido brevemente en el Capítulo 3), es un sistema operativo que se ejecuta en los motes Cross-Bow y cuyas aplicaciones típicas constan de diversos componentes- elementos de sensado, acceso de datos, etcétera-. Si la aplicación en cuestión usa una red en malla, como es el caso de este proyecto, debe incluirse en ella el componente del software *MultiHopRouter*. La implementación del componente *Multi-Hop* se encuentra en el directorio de contribuciones de Cross-Bow (proporcionado junto con el Kit de aplicación MOTE-KIT5040), y proporciona diversas mejoras sobre los algoritmos de redes en malla tradicionales.

El componente de ruteo *Multi-Hop*, transmite automáticamente cálculos estimado sobre la calidad de los enlaces, publica distancias estimadas entre nodos, ejecuta la selección de rutas óptimas y reenvía el tráfico de datos multi-saltos. Éste componente se divide en dos elementos principales: *MultiHopEngineM* y *MultiHopLEPSM*. Adicionalmente utiliza los componentes: *QueueSend*, *TimerC* y *GenericCommPromiscuous*; para ejecutar adecuadamente sus funciones.

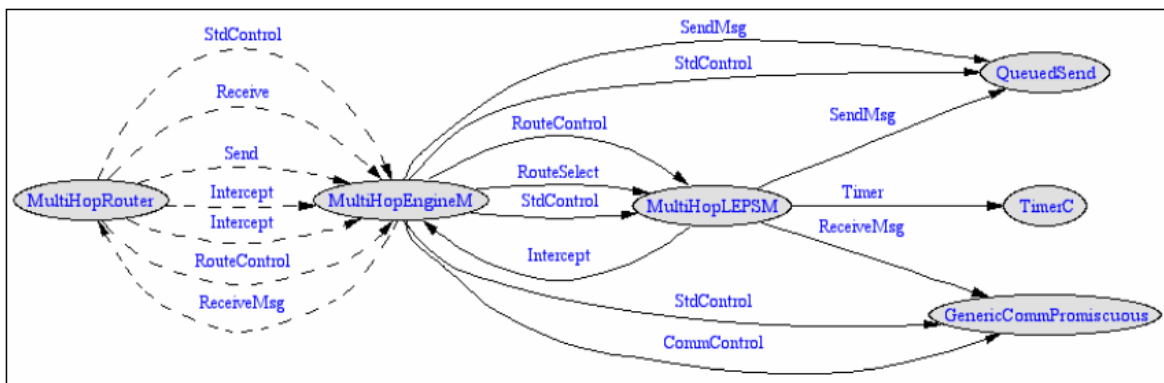


Figura A.1: Representación gráfica de la aplicación de TinyOS *Surge_Reliable* y *Surge_Reliable_Dot*

El componente *MultiHopEngineM* implementa el núcleo de reenvío y la función de envío del ruteo multi-saltos. Una vez que el nodo padre ha sido determinado, el componente *MultiHopEngineM* recibe, actualiza y transmite el paquete de datos hacia su destino final.

Como recibo del paquete Multi-saltos, el componente *MultiHopEngineM* reenvía el paquete si y solo si, éste fue directamente direccionado hacia el nodo vía el mecanismo

de direccionamiento de paquetes. Si es así, el *MultiHopEngineM* utiliza al *MultiHopLEPSM* para seleccionar el siguiente salto de destino y actualizar el paquete utilizando la Interfase de selección de ruta (*RouteSelect*). Una vez que el *MultiHopLEPSM* ha actualizado los archivos del paquete multi-saltos, el paquete es entregado al componente *QueueSend* para su pronta transmisión. El *QueueSend* es un componente muy simple, se encarga de aceptar múltiples peticiones de transmisión y efectuar las transmisiones actuales hasta donde el canal lo permita.

El movimiento de datos y las decisiones de ruteo del componente *MultiHopEngineM* son desintegradas en componentes separados con una sola interfase entre ellos para permitir que otros esquemas de decisión de ruteo sean fácilmente integrados en un futuro. El uso del *MultiHopRouter* es esencialmente transparente a las aplicaciones (mientras que utilicen correctamente la interfaz).

El uso del componente de librería Multi-Hop es en su mayor parte transparente a la aplicación. Cualquier aplicación que utilice la interfase *Send* puede ser conectada a este componente para llevar acabo la funcionalidad Multi-Hop. Si embargo, una de las limitaciones de Multi-Hop es el incremento en la tasa de datos. Las aplicaciones deben de mantener la frecuencia de mensajes promedio en un valor igual o menor a un mensaje cada dos segundos. Tasas de transmisión mayores a ésta pueden congestionar y/o desbordar el canal de comunicación.

Apéndice B: Componentes del Programa Surge_Reliable

```

SurgeM.nc           // Módulo Principal

includes Surge;
includes SurgeCmd;

/*
 * Data gather application
 */

module SurgeM {
  provides {
    interface StdControl;
  }
  uses {
    interface ADC as Batt;
    interface ADC as Temp;
    interface ADC as Light;
    interface ADCControl;
    interface Timer as TempTimer;
    interface Timer;
    interface Leds;
    interface StdControl as AccelCtl;
    interface ADC as AccelX;
    interface ADC as AccelY;
    interface StdControl as Sounder;
    interface StdControl as TempStdControl;
    interface StdControl as LightStdControl;
    interface Send;
    interface Receive as Bcast;
    interface RouteControl;
    interface AttrUse;
  }
}

implementation {

  enum {
    TIMER_GETADC_COUNT = 1,           // Timer ticks for ADC
    TIMER_CHIRP_COUNT = 10,         // Timer on/off chirp count
  };

  bool sleeping;                     // application command state
  bool focused;
  bool rebroadcast_adc_packet;

  IOS_Msg gMsgBuffer;
  norace uint16_t gSensorData;       // protected by gfSendBusy flag
  norace uint16_t gPowerData;       // protected by gfSendBusy flag
  norace uint16_t gTemp;             // protected by gfSendBusy flag
  norace uint16_t gAccelx;          // protected by gfSendBusy flag
  norace uint16_t gAccely;          // protected by gfSendBusy flag
  norace uint16_t gLight;           // protected by gfSendBusy flag
  bool gfSendBusy;

  uint32_t timer_rate;
  uint16_t timer_ticks;
}

```

```

/*****
* Initialization
*****/

static void initialize() {
#ifdef PLATFORM_PC
    outp(0x00, DDRF);
#endif
    timer_rate = INITIAL_TIMER_RATE - (TOS_LOCAL_ADDRESS << 3);
    atomic gfSendBusy = FALSE;
    sleeping = FALSE;
    rebroadcast_adc_packet = FALSE;
    focused = FALSE;
    call TempStdControl.init();
    call LightStdControl.init();
}

task void SendData() {
    SurgeMsg *pReading;
    uint16_t magx, magy;
    uint16_t Len;
    uint32_t batt;
    uint16_t error_no;
    dbg(DBG_USR1, "SurgeM: Sending sensor reading\n");

    if (pReading = (SurgeMsg *)call Send.getBuffer(&gMsgBuffer, &Len)) {
        pReading->type = SURGE_TYPE_SENSORREADING;
        pReading->parentaddr = call RouteControl.getParent();
        pReading->reading = gPowerData;
        pReading->seq_no ++;
        batt = ((uint32_t)(gSensorData)) << 23;
        pReading->seq_no &= 0x7ffff;
        pReading->seq_no += batt;
        pReading->temp = gTemp;
        pReading->light = gLight;
        pReading->accelx = gAccelx >> 2;
        pReading->accely = gAccely >> 2;
        call AttrUse.getAttrValue("mag_x", &magx, &error_no);
        call AttrUse.getAttrValue("mag_y", &magy, &error_no);
        pReading->magx = magx;
        pReading->magy = magy;
        gTemp = gLight = 0;

        if ((call Send.send(&gMsgBuffer, sizeof(SurgeMsg))) != SUCCESS)
            atomic gfSendBusy = FALSE;
    }
}

command result_t StdControl.init() {
    initialize();
    return SUCCESS;
}

command result_t StdControl.start() {
    call Timer.start(TIMER_REPEAT, timer_rate);

    call AttrUse.startAttr((call AttrUse.getAttr("mag_x"))->id);
    call AttrUse.startAttr((call AttrUse.getAttr("mag_y"))->id);

    return SUCCESS;
}

command result_t StdControl.stop() {
    return call Timer.stop();
}

```

```

/*****
* Commands and events
*****/

event result_t Timer.fired() {
    dbg(DBG_USR1, "SurgeM: Timer fired\n");
    timer_ticks++;
    if (timer_ticks % TIMER_GETADC_COUNT == 0) {
        //TOSH_SET_BAT_MON_PIN();
        //TOSH_uwait(250);
        call LightStdControl.start();
        call AccelCtl.start();
        call Light.getData();
    }
    // If we're the focused node, chirp
    if (focused && timer_ticks % TIMER_CHIRP_COUNT == 0) {
        call Sounder.start();
    }
    // If we're the focused node, chirp
    if (focused && timer_ticks % TIMER_CHIRP_COUNT == 1) {
        call Sounder.stop();
    }
    return SUCCESS;
}

async event result_t Light.dataReady(uint16_t data) {
    if (gLight == 0) {
        gLight = 1;
        call Light.getData();
        return SUCCESS;
    }
    gLight = data >> 2;
    call LightStdControl.stop();
    call TempStdControl.start();
    call TempTimer.start(TIMER_ONE_SHOT, 10);
    return SUCCESS;
}

event result_t TempTimer.fired() {
    call Temp.getData();
}

async event result_t Temp.dataReady(uint16_t data) {
    if (gTemp == 0) {
        gTemp = 1;
        call Temp.getData();
        return SUCCESS;
    }
    gTemp = data >> 2;
    call TempStdControl.stop();
    call AccelX.getData();
    return SUCCESS;
}

async event result_t AccelX.dataReady(uint16_t data) {
    gAccelx = data;
    call AccelY.getData();
    return SUCCESS;
}

async event result_t AccelY.dataReady(uint16_t data) {
    gAccely = data;
    call Batt.getData();
    return SUCCESS;
}

```

```

async event result_t Batt.dataReady(uint16_t data) {
    //SurgeMsg *pReading;
    //uint16_t Len;
    //TOSH_CLR_BAT_MON_PIN();
    dbg(DBG_USR1, "SurgeM: Got ADC reading: 0x%x\n", data);
    atomic {
        if (!gfSendBusy) {
            gfSendBusy = TRUE;
            gSensorData = data;
            post SendData();
        }
    }
    return SUCCESS;
}

event result_t Send.sendDone(TOS_MsgPtr pMsg, result_t success) {
    dbg(DBG_USR2, "SurgeM: output complete 0x%x\n", success);
    //call Leds.greenToggle();
    atomic gfSendBusy = FALSE;
    return SUCCESS;
}

/* Command interpreter for broadcasts
 *
 */

event TOS_MsgPtr Bcast.receive(TOS_MsgPtr pMsg, void* payload, uint16_t payloadLen) {
    SurgeCmdMsg *pCmdMsg = (SurgeCmdMsg *)payload;

    dbg(DBG_USR2, "SurgeM: Bcast type 0x%02x\n", pCmdMsg->type);

    if (pCmdMsg->type == SURGE_TYPE_SETRATE) { // Set timer rate
        timer_rate = pCmdMsg->args.newrate;
        dbg(DBG_USR2, "SurgeM: set rate %d\n", timer_rate);
        call Timer.stop();
        call Timer.start(TIMER_REPEAT, timer_rate);
    } else if (pCmdMsg->type == SURGE_TYPE_SLEEP) {
        // Go to sleep - ignore everything until a SURGE_TYPE_WAKEUP
        dbg(DBG_USR2, "SurgeM: sleep\n");
        sleeping = TRUE;
        call Timer.stop();
        call Leds.greenOff();
        call Leds.yellowOff();
    } else if (pCmdMsg->type == SURGE_TYPE_WAKEUP) {
        dbg(DBG_USR2, "SurgeM: wakeup\n");

        // Wake up from sleep state
        if (sleeping) {
            initialize();
            call Timer.start(TIMER_REPEAT, timer_rate);
            sleeping = FALSE;
        }
    }
}

```

```

async event result_t Batt.dataReady(uint16_t data) {
    //SurgeMsg *pReading;
    //uint16_t Len;
    //TOSH_CLR_BAT_MON_PIN();
    dbg(DBG_USR1, "SurgeM: Got ADC reading: 0x%x\n", data);
    atomic {
        }
    } else if (pCmdMsg->type == SURGE_TYPE_FOCUS) {
    dbg(DBG_USR2, "SurgeM: focus %d\n", pCmdMsg->args.focusaddr);
    // Cause just one node to chirp and increase its sample rate;
    // all other nodes stop sending samples (for demo)
    if (pCmdMsg->args.focusaddr == TOS_LOCAL_ADDRESS) {
        // OK, we're focusing on me
        focused = TRUE;
        call Sounder.init();
        call Timer.stop();
        call Timer.start(TIMER_REPEAT, FOCUS_TIMER_RATE);
    } else {
        // Focusing on someone else
        call Timer.stop();
        call Timer.start(TIMER_REPEAT, FOCUS_NOTME_TIMER_RATE);
    }
    }
    } else if (pCmdMsg->type == SURGE_TYPE_UNFOCUS) {
    // Return to normal after focus command
    dbg(DBG_USR2, "SurgeM: unfocus\n");
    focused = FALSE;
    call Sounder.stop();
    call Timer.stop();
    call Timer.start(TIMER_REPEAT, timer_rate);
    }
    return pMsg;
}

event result_t AttrUse.getAttrDone(char *name, char *resultBuf, SchemaErrorNo
errorNo) {
    return SUCCESS;
}

event result_t AttrUse.startAttrDone(uint8_t id)
{
    return SUCCESS;
}

}

call Leds.yellowOff();

} else if (pCmdMsg->type == SURGE_TYPE_WAKEUP) {
    dbg(DBG_USR2, "SurgeM: wakeup\n");

    // Wake up from sleep state
    if (sleeping) {
        initialize();
        call Timer.start(TIMER_REPEAT, timer_rate);
        sleeping = FALSE;
    }
}

```

Los componentes incluidos por el programa anterior son: Surge.nc, Surge.h y SurgeCmd.h. A su vez Surge.nc incluye la aplicación Multi-Hop.

Surge.nc *// Interfase*

```

includes Surge;
includes SurgeCmd;
includes MultiHop;

configuration Surge {
}
implementation {
    components Main, SurgeM, TimerC, LedsC, NoLeds, ADCC, RandomLFSR, PhotoTemp,
        GenericCommPromiscuous as Comm, Bcast, EWMA MultiHopRouter as multihopM, Que
uedSend, Sounder, Accel, AttrMag, Attr;

    Main.StdControl -> SurgeM.StdControl;
    Main.StdControl -> Bcast.StdControl;
    Main.StdControl -> multihopM.StdControl;
    Main.StdControl -> QueuedSend.StdControl;
    Main.StdControl -> TimerC;
    Main.StdControl -> Comm;
    Main.StdControl -> AttrMag;
    // multihopM.CommControl -> Comm;

    SurgeM.ADCControl -> ADCC;
#ifdef PLATFORM_PC
    SurgeM.Batt -> ADCC.ADC[IOS_ADC_BANDGAP_PORT];
#else
    SurgeM.Batt -> ADCC.ADC[10];
#endif
    SurgeM.Temp -> PhotoTemp.ExternalTempADC;
    SurgeM.Light -> PhotoTemp.ExternalPhotoADC;
    SurgeM.AccelX -> Accel.AccelX;
    SurgeM.AccelY -> Accel.AccelY;
    SurgeM.AccelCtl -> Accel;
    SurgeM.TempStdControl -> PhotoTemp.TempStdControl;
    SurgeM.LightStdControl -> PhotoTemp.PhotoStdControl;
    SurgeM.TempTimer -> TimerC.Timer[unique<"Timer">];
    SurgeM.Timer -> TimerC.Timer[unique<"Timer">];
    SurgeM.Leds -> LedsC; // NoLeds;
    SurgeM.Sounder -> Sounder;
    SurgeM.AttrUse -> Attr.AttrUse;

    SurgeM.Bcast -> Bcast.Receive[AM_SURGECMDMSG];
    Bcast.ReceiveMsg[AM_SURGECMDMSG] -> Comm.ReceiveMsg[AM_SURGECMDMSG];

    SurgeM.RouteControl -> multihopM;
    SurgeM.Send -> multihopM.Send[AM_SURGEMSG];
    multihopM.ReceiveMsg[AM_SURGEMSG] -> Comm.ReceiveMsg[AM_SURGEMSG];
    //multihopM.ReceiveMsg[AM_MULTIHOPMSG] -> Comm.ReceiveMsg[AM_MULTIHOPMSG];
}

```

Surge.h

```

enum{
    INITIAL_TIMER_RATE = 1024 * 8,
    FOCUS_TIMER_RATE = 1000,
    FOCUS_NOIIME_TIMER_RATE = 1000
};

enum {
    SURGE_TYPE_SENSORREADING = 0,
    SURGE_TYPE_ROOTBEACON = 1,
    SURGE_TYPE_SETRATE = 2,
    SURGE_TYPE_SLEEP = 3,
    SURGE_TYPE_WAKEUP = 4,
    SURGE_TYPE_FOCUS = 5,
    SURGE_TYPE_UNFOCUS = 6
};

typedef struct SurgeMsg {
    uint8_t type;
    uint16_t reading;
    uint16_t parentaddr;
    uint32_t seq_no;
    uint8_t light;
    uint8_t temp;
    uint8_t magx;
    uint8_t magy;
    uint8_t accelx;
    uint8_t accely;
} __attribute__((packed)) SurgeMsg;

enum {
    AM_SURGEMSG = 17
};

```

SurgeCmd.h // *Fichero de Configuración*

```

typedef struct SurgeCmdMsg {
    uint8_t type;
    union {
        // FOR SURGE_TYPE_SETRATE
        uint32_t newrate;
        // FOR SURGE_TYPE_FOCUS
        uint16_t focusaddr;
    } args;
} __attribute__((packed)) SurgeCmdMsg;

enum {
    AM_SURGECMDMSG = 18
};

```