



INSTITUTO TECNOLÓGICO DE MEXICALI

“Uso de Redes de Sensores Inalámbricas y  
Métodos de Fusión de Información para la  
Detección de Incendios Forestales”

Presenta

Luis Aram Tafoya Díaz

Tesis

Para obtener el grado de

Maestro en Ingeniería Electrónica

Director de tesis

Dr. Arnoldo Díaz Ramírez



Mexicali, B.C.

Diciembre 2011

# Agradecimientos

Quiero agradecer a todas aquellas personas que han hecho posible la terminación del escrito. Estoy muy agradecido con Dr. Arnoldo Díaz por el asesoramiento brindado a lo largo de la investigación. A mi madre por comprender y ayudarme a tener un buen desempeño diariamente. A mis hermanos por ser parte de esta etapa de mi vida. A mi padre por la educación y formación que me brindó.

# Resumen

En los últimos años las redes inalámbricas MANET han recibido gran atención por parte de la comunidad científica. Estas redes se caracterizan por no requerir de infraestructura fija. Un caso especial de las redes MANET son las redes de sensores inalámbricas o WSN, que están formadas por nodos capaces de monitorizar el medio ambiente y de comunicarse vía inalámbrica. Para poder aprovechar la funcionalidad de las redes de sensores inalámbricas, sobre todo tomando en consideración los recursos limitados de los nodos, es necesario contar con una plataforma de software que optimice los recursos y que facilite el desarrollo de aplicaciones.

Una de las áreas de aplicación de las WSN que más atención ha recibido recientemente es la detección de eventos. Uno de los eventos que causa enormes pérdidas económicas y materiales son los incendios. Por lo cual se propone y evalúan dos métodos para la detección de incendios en bosques. Los métodos utilizan la fusión de información. Están diseñados para implementarse en una red que contenga sensores de temperatura, humedad e iluminación para el método basado en umbrales y con sensores temperatura y humedad para el método basado en Dempster-Shafer. Para evaluar los métodos se registraron mediciones por varios días y se generaron artificialmente algunos incendios. Los resultados obtenidos de la evaluación realizada en los experimentos demuestran que los métodos son capaces de detectar eficientemente los incendios existentes.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos Generales . . . . .	1
1.2. Objetivos Específicos . . . . .	1
1.3. Motivación . . . . .	2
1.4. Organización del documento . . . . .	3
<b>2. Redes Inalámbricas</b>	<b>4</b>
2.1. Red de Computadoras . . . . .	4
2.2. Red Inalámbrica . . . . .	5
2.3. Redes MANET y Mesh . . . . .	5
2.3.1. Estándar IEEE 802.11 . . . . .	8
2.3.2. Estándar IEEE 802.15 . . . . .	9
<b>3. Redes de Sensores Inalámbricas</b>	<b>11</b>
3.1. Definición . . . . .	11
3.2. Estándar IEEE 802.15.4 . . . . .	12
3.3. Desarrollo a la fecha . . . . .	17
3.3.1. Técnicas para áreas residenciales . . . . .	17
3.3.2. Técnicas para eventos en bosques . . . . .	21
3.3.3. Técnicas para detección temprana . . . . .	32
3.4. Herramientas de Desarrollo . . . . .	36
3.5. Plataformas . . . . .	38
<b>4. TinyOS y NesC</b>	<b>44</b>
4.1. TinyOS . . . . .	44
4.2. nesC . . . . .	45
4.3. Instalación y Configuración . . . . .	46
<b>5. Métodos propuestos para la detección de incendios forestales</b>	<b>48</b>
5.1. Algoritmo para la detección de fuego basado en umbrales . . . . .	48
5.2. Teoría de fusión de datos Dempster-Shafer . . . . .	52
5.3. Método basado en la teoría Dempster-Shafer para la detección de fuego . . . . .	54
5.4. Interpolación Lineal . . . . .	54

## ÍNDICE GENERAL

---

5.5. Interpolación de Lagrange . . . . .	54
5.6. Formulación de Newton con diferencias divididas . . . . .	55
5.7. Diagrama de flujo para la detección de fuego . . . . .	56
5.8. Cálculo de Masas . . . . .	56
5.9. Fusión de Masas y Normalización . . . . .	60
5.10. Ejemplo del Método de Detección . . . . .	60
<b>6. Evaluación</b>	<b>65</b>
6.1. Algoritmo para la detección de fuego basado en umbrales . . . .	66
6.2. Método para detección de Fuego basado en Dempster-Shafer . . .	67
<b>7. Conclusiones y Trabajo Futuro</b>	<b>74</b>

# Índice de cuadros

2.1. Estándares . . . . .	8
3.1. Formato del paquete para CC1100 en bits[57] . . . . .	17
3.2. Formato del paquete[5] . . . . .	20
5.1. Representación de los superíndices . . . . .	56
5.2. Representación de los subíndices . . . . .	57
5.3. Estados . . . . .	58
5.4. Masa probabilística individual . . . . .	60
5.5. Asignación de Masas . . . . .	60
5.6. Cálculos de fusión y normalización . . . . .	64
6.1. Resultados de algoritmo basado en umbrales . . . . .	66
6.2. Resultados del método basado en Dempster-Shafer . . . . .	73

# Índice de figuras

2.1. Ejemplo de MANET . . . . .	6
3.1. Ejemplo de WSN . . . . .	12
3.2. Topología de Estrella [38] . . . . .	14
3.3. Topología Peer-Peer [38] . . . . .	14
3.4. Adaptación de Modelo OSI en IEEE 802.15.4 . . . . .	15
3.5. Arquitectura de la red [57] . . . . .	18
3.6. Resultado de los Experimentos [56] . . . . .	20
3.7. Una red de sensores inalámbrica para la detección de incendios forestales en tiempo real [30] . . . . .	21
3.8. Procesamiento de Paquete RR usando el método de red neuronal [30] . . . . .	22
3.9. Estructura del Sistema FWI [32] . . . . .	23
3.10. Máquina de Estados para definir las condiciones del ambiente [44] . . . . .	24
3.11. Arquitectura del sistema de monitoreo [12] . . . . .	25
3.12. La arquitectura de FireNet [26] . . . . .	26
3.13. Arquitectura de LACU [13] . . . . .	27
3.14. Estructura del sistema [53] . . . . .	28
3.15. Transmisión de datos [25] . . . . .	30
3.16. Estructura de los componentes de FFSS [9] . . . . .	31
3.17. Resultados de los experimentos [28] . . . . .	32
3.18. Mensajes de aviso con diferente numero de nodos vecinos defectuosos [55] . . . . .	33
3.19. Estructura de las tres capas del sistema para la detección de fuego [50] . . . . .	34
3.20. Resultados por el caso de un sensor con $c = 0,01$ , $v = 1,0$ , y $L_a = L_1 = 1,0$ a) Function Infinite-horizon b) Optimum stationary LDF thresholds [54] . . . . .	40
3.21. Resultados por el caso de dos sensores idénticos con $c = 0,01$ , $v = 1,0$ , y $L_a = L_1 = 1,0$ a) Function Infinite-horizon b) Optimum stationary LDF thresholds [54] . . . . .	41
3.22. Estructura del sistema [46] . . . . .	42
3.23. Ejemplo de configuración de una WSN [48] . . . . .	42
3.24. Respuesta de la temperatura [19] . . . . .	43

## NOMENCLATURA

---

5.1. Generación artificial de incendio . . . . .	49
5.2. Mediciones en condiciones normales a)Temperatura, b)Luz visible, y c)Humedad relativa . . . . .	49
5.3. Máquina de estados utilizada por el algoritmo . . . . .	51
5.4. Diagrama De Flujo para la Detección de Fuego . . . . .	57
6.1. Formato Paquete de Datos . . . . .	65
6.2. Formato Tabla MUESTRAS . . . . .	66
6.3. Experimento de 9:47 a 9:52 horas . . . . .	68
6.4. Experimento de 9:59 a 10:04 horas . . . . .	68
6.5. Experimento de 12:22 a 12:27 horas . . . . .	69
6.6. Experimento de 12:38 a 12:42 horas . . . . .	69
6.7. Experimento de 14:49 a 14:55 horas . . . . .	70
6.8. Experimento de 15:33 a 15:39 horas . . . . .	70
6.9. Experimento de 17:21 a 17:27 horas . . . . .	71
6.10. Experimento de 17:38 a 17:46 horas . . . . .	71
6.11. Experimento de 18:59 a 19:05 horas . . . . .	72
6.12. Experimentos con fuego . . . . .	72



# Nomenclatura

6LoWPAN	IPv6-based Low power Wireless Personal Area Networks
ACK	Acknowledgement
AWMA	Alternating Wireless Medium Access
BSD	Berkeley Software Distribution
DQDB	Distributed queue dual bus
FFD	Full Function Device
FTP	File Transfer Protocol
FWI	FIRE Weather Index
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HGA	Hardware-Based Genetic Algorithm
LAN	Local Area Network
LR-WPAN	low rate Wireless Personal Area Network
MAC	Medium Acces Control
MAN	Metropolitan Area Network
PC	Personal Computer
PDA	Personal Digital Assistant
PIB	PAN information base
PID	Proportional–Integral–Derivative
PSTN	Public Switched Telephone Network
RFD	Reduced Function Device

## NOMENCLATURA

---

RSSI	Receive Signal Strength Indication
UART	Universal Asynchronous Receiver/Transmitter
VMBA	Voting Median Base Algorithm for Approximate performance Measurements of Wireless Sensor Networks
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network

# Capítulo 1

## Introducción

Las redes inalámbricas especializadas (ad hoc) permite a un grupo de nodos de la red la posibilidad de autoconfigurarse y mantener la red de manera autónoma. Gracias a esta posibilidad, resultan útiles en situaciones donde es requerida una respuesta rápida de un evento en escenarios donde no es posible contar con infraestructura fija. Dentro de las redes ad hoc se encuentran las redes de sensores inalámbricas, lo interesante de estas redes es que tienen la capacidad de medir condiciones en el entorno físico. Las redes de sensores inalámbricas están compuestas por un gran numero de nodos sensores, también llamados motes, que básicamente cuentan con procesador, memoria, sensor, radio de comunicación y fuente de energía. Debido a la falta de infraestructura este tipo de red, hace que cada mote se configure, se administre y se controle de forma autónoma, contando con funciones de encaminamiento y dirección de datos. Para el funcionamiento de los nodos se utilizan protocolos aplicables a la arquitectura de las redes de sensores inalámbricas. Debido a que los motes cuentan con capacidad limitada el sistema operativo debe mostrar ser eficiente con pocos recursos. Las aplicaciones realizadas para los motes que utilizan TinyOS [3] se programan en lenguaje nesC [7]. Para la detección de eventos es posible utilizar la fusión de información, se encuentran disponibles varios métodos, los utilizados fueron el método basado en umbrales y el método Dempster-Shafer.

### 1.1. Objetivos Generales

Diseñar una metodología con la cual sea posible detectar a tiempo la presencia de fuego en bosques, utilizando las redes de sensores inalámbricas. La cual sera evaluada.

### 1.2. Objetivos Específicos

- Utilizar las Redes de Sensores Inalámbricas para medir las condiciones del ambiente.

### 1.3. MOTIVACIÓN

---

- Identificar las condiciones ambientales normales.
- Identificar el comportamiento del fuego en el ambiente.
- Diseñar métodos para detectar fuego en los bosques.
- Probar los métodos planteados, mostrando los resultados obtenidos.

### 1.3. Motivación

Las redes de sensores tienen un gran potencial para utilizarse en diversas aplicaciones, tales como rastreo de objetivos militares, vigilancia, detección de eventos, recuperación después desastres naturales, control de la salud, exploración en zonas de alto riesgo o detección de sismos, por mencionar algunas [24].

Debido a las limitaciones en memoria y capacidad de procesamiento, los nodos cuentan con un radio para transmitir de manera inalámbrica únicamente la información requerida y parcialmente procesada. Esta información es enviada hacia una estación base, como por ejemplo, una computadora portátil o una tableta electrónica. La estación base recibe la información de los nodos, la procesa y lleva a cabo las acciones necesarias, como por ejemplo, informar a los usuarios si algún evento de interés ha ocurrido. Generalmente la fuente primaria de energía de los nodos son baterías, y en algunos casos del medio ambiente se obtiene una segunda fuente de energía (por ejemplo, utilizando paneles solares). Por otra parte, los actuadores pueden ser incorporados a los nodos dependiendo del tipo de sensores y de la aplicación.

Una de las aplicaciones para redes de sensores que mayores beneficios puede proporcionar es la detección de incendios en bosques. Tan solo en el 2010, en el estado de California (Estados Unidos) se registraron aproximadamente 5,000 incendios forestales. En ese mismo estado, el promedio de incendios de los últimos cinco años es de 4,972 incendios por año, con pérdidas económicas anuales estimadas en \$ 292,962,667 dólares americanos, y una importante cantidad de personas afectadas [43].

Existen diversos métodos para monitorizar incendios forestales. Un método tradicional y que aún se utiliza consiste en revisiones visuales por parte de personas colocadas en torres de observación. Desafortunadamente, este método tiene limitaciones y esta expuesto a errores humanos. Por esta razón es que se han propuesto técnicas basadas en el uso de la tecnología, como el Sistema Automático Avanzado de Vídeo Vigilancia [15] o en el uso de imágenes obtenidas por satélites [39]. Sin embargo, la baja resolución espacial y temporal de las imágenes obtenidas por los satélites puede retardar la detección de los incendios, de tal manera que cuando éstos sean detectados ya se hayan propagado demasiado.

Las aplicaciones de detección de eventos para redes de sensores son esencialmente procesos de fusión de información. Es por ello que surge el interés de

diseñar métodos para la detección de incendios forestales utilizando la fusión de información.

### 1.4. Organización del documento

El documento se encuentra organizado en siete capítulos en el capítulo 1 se mencionan los objetivos generales y específicos primordiales para la realización de la investigación, así como la motivación que demuestra el interés por el diseño de métodos. En el capítulo 2 se describe a las redes de computadoras, se ven los diferentes tipos de red posibles en las redes inalámbricas, así como los estándares presentes en las redes inalámbricas. En el capítulo 3 se define la WSN (Wireless Sensor Network) y se describe el desarrollo a la fecha relacionado con la detección de fuegos en áreas residenciales, en bosques y utilizando detección temprana. En el capítulo 4 se menciona el sistema operativo utilizado en las WSN así como el lenguaje de programación para la realización de aplicaciones, se indican los pasos necesarios para la instalación y configuración de TinyOS. En el capítulo 5 se describen los dos métodos propuestos el primero basado en umbrales que utiliza sensor de temperatura, humedad relativa y luz visible, el segundo se basa en el método Dempster-Shafer utiliza el sensor de temperatura y humedad relativa. En el capítulo 6 se muestran los resultados obtenidos con la evaluación planteada. En el capítulo 7 se menciona las conclusiones y trabajo futuro relacionado con los métodos propuestos.

## Capítulo 2

# Redes Inalámbricas

### 2.1. Red de Computadoras

Una red esta compuesta por un conjunto de hosts que se comunican entre si por medio de cables, ondas o señales. Un host es una computadora o dispositivo que puede proveer información, recursos y/o servicios.

La red de comunicaciones esta compuesta por un conjunto de medios técnicos que permiten la comunicación entre equipos de computo autónomos. Por lo general, se transmiten datos, audio y vídeo por medio de cable, fibra óptica, aire, entre otros. Para la comunicación entre programas de distintos equipos de computo, la ISO especifico el Modelo de Referencia OSI en siete capas, las cuales desarrollan una función específica.

Una LAN (Local Area Network) es una red de propiedad privada ubicada en un área de pocos kilómetros, por ejemplo dentro de un edificio o una universidad. Se utilizan para compartir recursos como impresoras, escáner, además para intercambiar archivos entre computadoras personales y estaciones de trabajo, se ven mucho este tipo de redes en oficinas y fabricas. La LAN debido a su pequeño tamaño, simplifica su administración, permite probar distintas topologías.

Una MAN es muy similar a la red LAN en cuanto a su tecnología, solo que puede extenderse en un área mayor, por ejemplo, puede abarcar un grupo de campus ubicados en una ciudad, la cual puede ser privada o publica. La MAN pueden manejar datos, voz y vídeo. La característica que distingue a una MAN es que utiliza el estándar DQDB. El DQDB consiste en dos cables unidireccionales, los cuales están conectados a todas las computadoras.

Una WAN se extiende en una área mucho mayor, puede ser un país o un continente. Las WAN están compuestas por un conjunto de subredes, las cuales están compuestas por líneas de transmisión y por elementos de conmutación. Las líneas de transmisión se encargan de enviar bits de un host a otro. Los elementos de conmutación mejor conocidos como enrutadores se encargan de recibir los datos por un línea de entrada y escoger una línea de salida para reenviarlos. La subredes están compuestas por la colección de líneas de comunicación y los

## 2.2. RED INALÁMBRICA

---

enrutadores.

## 2.2. Red Inalámbrica

Las redes inalámbricas tienen muchos usos. Son utilizados en diferentes lugares ya sea en tierra, aire o mar. La gente las utiliza para realizar llamadas con su teléfono celular en el cual se puede hacer llamadas y navegar por internet a la vez. Tienen buenas aplicaciones uno de ellos es en sitios de desastre como incendios, inundaciones, temblores donde se ha dañado la comunicación por cable. Se puede llegar a contar con una red propia, contando con el equipo de recepción inalámbrica adecuada.

Las redes inalámbricas tienen muchas formas, la gran mayoría de las universidades ya cuentan con antenas para brindar acceso a internet, con lo que los alumnos pueden revisar sus correos, navegar, consultar tarjetas del catalogo de la biblioteca, etc.

## 2.3. Redes MANET y Mesh

Las redes MANET están compuestas por un conjunto de nodos móviles inalámbricos, capaces de auto-organizarse dinámicamente utilizando diversas topologías de red [10]. Una de las características mas interesantes de este tipo de redes consiste en que no requieren de una infraestructura fija, por lo que cada nodo funciona como encaminador al retransmitir paquetes hasta el nodo destino. La Fig. 2.1 muestra un ejemplo de una red MANET formada por diversos dispositivos móviles, que pueden comunicarse entre sí sin necesidad de un nodo coordinador o punto de acceso. Con redes de este tipo se puede compartir información en lugares en donde no existe infraestructura fija o en los que ha ocurrido algún desastre y se ha destruido la infraestructura existente, permitiendo el rápido establecimiento de una red de comunicación.

De acuerdo a sus características los protocolos pueden clasificarse en:

1. Proactivos, Reactivos, Híbridos
2. Jerárquicos, Planos
3. Geográficos, No geográficos
4. Encaminamiento en el origen, encaminamiento salto a salto
5. Singlepath, multipath

A continuación se describen las clasificaciones:

- **Proactivos.** Este tipo de protocolo constantemente está enviando información sobre el encaminamiento para que cualquier nodo pueda comunicarse con otro dentro de la red, esta característica agiliza la respuesta a solicitudes de ruta ofreciendo buen comportamiento en situaciones donde

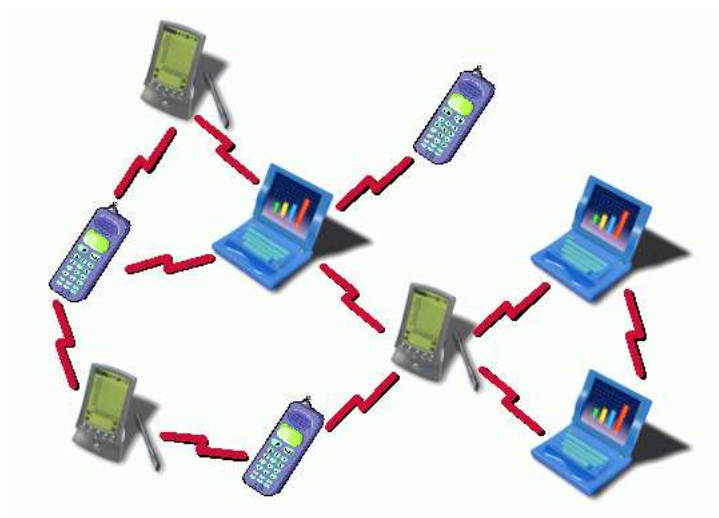


Figura 2.1: Ejemplo de MANET

la movilidad de los nodos es muy alta. Entre estos protocolos se encuentran DSDV (Destination-Sequenced Distance Vector), WRP (Wireless Routing Protocol), CGSR (Clusterhead Gateway Switch Routing).

- **Reactivos.** Los protocolos reactivos tienen un sobre carga menor debido a que crean rutas solo cuando es necesario, tienen retrasos cuando el establecimiento de rutas son muy constantes. Dentro de estos protocolos se encuentran AODV (Ad Hoc On-Demand Distance Vector), DSR (Dynamic Source Routing), TORA (Temporally-Ordered Routing Algorithm), ABR (Associativity-Based Routing), CBRP (Cluster Based Routing Protocol), RDMAR (Relative-Distance Micro-discovery Ad hoc Routing protocol).
- **Híbridos.** Se mantiene la filosofía proactiva en un ámbito local y reactiva a nivel global, dentro de estos protocolos se encuentra ZRP (Zone Routing Protocol).
- **Jerárquico.** En los protocolos jerárquicos los nodos se clasifican en niveles y su retransmisión depende del nivel en el que este. Uno de los protocolos jerárquicos más estables es CGSR(Clusterhead Gateway Switch Routing).
- **Plano.** En los planos todos los nodos tienen el mismo nivel y las mismas funciones y responsabilidades.
- **Geográficos.** En los protocolos geográficos se toma en cuenta su posición geográfica exacta de cada nodo para realizar los encaminamientos, el inconveniente que tiene es que el dispositivo debe tener un chip para posicionamiento vía GPS (Global Positioning System). Uno de los pro-



### 2.3. REDES MANET Y MESH

---

protocolos relacionados es DREAM (Distance Routing Effect Algorithm for Mobility).

- **Encaminamiento en el origen.** En los protocolos de encaminamiento en el origen cada paquete que se envía lleva la ruta que deberá seguir para llegar a su destino por esta razón los nodos intermediarios reducen su procesamiento. Uno de los protocolos que utiliza encaminamiento en el origen es DSR (Dynamic Source Routing).
- **Encaminamiento salto a salto.** En los protocolos de encaminamiento salto a salto el nodo intermediario deberá decidir cuál será el nodo siguiente.
- **Singlepath.** En los protocolos singlepath solo se mantiene una ruta única hacia cada destino,
- **Multipath.** En los multipath se mantienen varias rutas hacia cada destino. Dentro de los protocolos multipath se encuentra TORA (Temporally-Ordered Routing Algorithm).

Un caso especial de las redes MANET son las redes de sensores inalámbricas o WSN (*Wireless Sensor Networks*), que están formadas por nodos capaces de monitorizar el medio ambiente y de comunicarse vía inalámbrica. Las WSN han mostrado un gran avance recientemente debido a que los sensores cada vez son mas pequeños, económicos e inteligentes. El diseño de una WSN depende significativamente del objetivo de la aplicación, el entorno donde se utilizarán y los costos que esto implica.

Las Wireless mesh networks<sup>1</sup> (WMNs) consisten de routers en malla y clientes en malla, donde los routers en malla tienen movilidad mínima y forman el backbone de las WMNs. Estos proveen acceso a la red para ambos en malla y clientes convencionales. La integración de las WMN con otras redes como el Internet, celular, IEEE 802.11, IEEE 802.15, IEEE 802.16, WSN, etc., puede ser realizado en la función del gateway y puente en los routers mesh. Los clientes mesh pueden anticiparse para resolver limitaciones y para mejorar significativamente el rendimiento de las redes ad hoc, WLANs, WPAN y WMAN. Se encuentran sometidos a un rápido progreso e inspiración de numerosos desarrolladores. Las WSN pueden entregar servicios inalámbricos para una larga variedad de aplicaciones ya sean personales, locales, en campus y áreas metropolitanas.

La arquitectura de las WMN puede ser clasificada en tres grandes grupos basados en la funcionalidad de los nodos [22]:

- **Infraestructura/Backbone WMNs.** Este tipo de WMNs incluye routers mesh formando una infraestructura para clientes que pueden conectarse. La WMN infraestructura/backbone puede construirse utilizando varios tipos de tecnologías de radio, la mas utilizada es la IEEE 802.11. Para clientes convencionales con la misma tecnología de radio que los routers

---

<sup>1</sup>Redes inalámbricas en malla

## 2.3. REDES MANET Y MESH

---

mesh, pueden comunicarse directamente con los routers mesh. Si se utiliza diferente tecnología de radio, los clientes pueden comunicarse con las estación base que tienen comunicación Ethernet hacia los routers mesh.

- **WMNs Cliente.** El cliente en malla provee una red peer-to-peer entre dispositivos cliente. En este tipo de arquitectura, los nodos cliente constituyen la red actual para mejorar el encaminamiento y funcionalidades de configuración, también proveen aplicaciones a los consumidores. Por lo tanto, no se necesita el router para este tipo de redes. En las WMN Cliente, un paquete destinado a un nodo en la red salta múltiples nodos para llegar a su destino. En las WSN Cliente usualmente son formados utilizando un tipo de radio en los dispositivos. Los requerimientos de los dispositivos del usuario final se incrementan cuando se compara con una infraestructura en malla.
- **WMN Híbridas.** Esta arquitectura es la combinación de la infraestructura y malla cliente. Los clientes malla pueden acceder a la red a través de routers mesh también pueden directamente con otros clientes en malla. Mientras la infraestructura provee conectividad con otras redes como Internet, Wi-Fi, WiMAX, celular y WSN; las capacidades de encaminamiento de los clientes proveen conectividad y cobertura dentro de la WMN.

### 2.3.1. Estándar IEEE 802.11

Wi-Fi es un estándar aprobado por la IEEE, también conocido como 802.11b. En el año 2004 se certificaron dos nuevas versiones la 802.11a y la 802.11g.

La tecnología 802.11g es compatible con las aplicaciones desarrolladas en 802.11b, se espera que pronto sea compatible con con el estándar 802.11a. Por lo que la tecnología 802.11g es la que soporta mas versiones.

La finalidad de la tecnología Wi-Fi es ser utilizada en redes locales inalámbricas, además permite acceder a Internet. La certificación de los equipos con esta tecnología es llevada a cabo por la organización comercial Wi-Fi Alliance.

El estándar 802.11b puede llegar a tener una velocidad de 11 Mbps en la banda de 2.4 GHz, mientras que el estándar 802.11g tiene una velocidad de 54 Mbps en la banda de 2.4 GHz, el estándar IEEE 802.11n en la banda de 2.4 GHz tiene una velocidad de 150 Mbps. Ver cuadro 2.1.

Estándar IEEE	Velocidad (Mbps)	Banda (GHz)
802.11a	54	5
802.11b	11	2.4
802.11g	54	2.4
802.11n	150	2.4

Cuadro 2.1: Estándares

El estándar IEEE 802.11a también conocido como WIFI 5 opera en la banda de 5 GHz, es utilizado en Estados Unidos y Japón, mientras en la Unión Europea

### 2.3. REDES MANET Y MESH

---

aún no es aprobado debido a que están considerando utilizar el estándar europeo HIPERLAN.

La tecnología inalámbrica Bluetooth funciona en la frecuencia de 2.4 GHz desde la versión 1.2 se mejoro para que no presente interferencias con la tecnología 802.11x.

La tecnología Wi-Fi cuenta con protocolos de seguridad de datos como son WEP y WPA, los cuales se encargan de la autenticación, integridad y confidencialidad. Actualmente el protocolo de seguridad mas recomendable es el WPA2 mejora del WPA.

#### 2.3.2. Estándar IEEE 802.15

El IEEE 802.15 se enfoca en el desarrollo de estándares para redes inalámbricas de área personal<sup>2</sup>. Las WPAN son utilizadas para redes inalámbricas para dispositivos móviles como computadoras, PDA, teléfonos celulares, entre otros dispositivos electrónicos.

##### IEEE 802.15.1 Bluetooth

Bluetooth es basado en radio frecuencia<sup>3</sup> de corto alcance para dispositivos portátiles. Funciona en la banda de frecuencia de 2.4 GHz a 720 Kb/s. Utiliza saltos de frecuencia<sup>4</sup>, permitiendo dividir la banda de frecuencia en 79 canales (2.402 - 2.480 GHz). El transceiver de radio permite saltar de un canal a otro de una manera pseudo-aleatorio, asignada por el dispositivo maestro. En una piconet soporta hasta 8 dispositivos. Las piconets puede combinarse para formar Scatternets.

##### IEEE 802.15.2 Coexistencia

El IEEE 802.15 desarrolla practicas para facilitar la coexistencia para WPAN y WLAN. Se cuantifica la interferencia mutua de una WLAN y una WPAN [2]. Se han creado mecanismos para facilitar la coexistencia entre dispositivos estandarizados y no estandarizados que utilizan la banda de 2.4 GHz para transmisiones inalámbricas en WLAN<sup>5</sup> y WPAN. Los mecanismos son:

- AWMA<sup>6</sup>: mecanismos para transmisiones 802.11 y bluetooth desde la misma laptop o PDA. El cual está administrado por la MAC. Las unidades se conectan y sincronizan al mismo punto de acceso, por lo cual no existirá interferencia.
- Packet Traffic Arbitration: son técnicas colaborativas que definen reglas de administración directamente entre los sistemas.

---

<sup>2</sup>(WPAN Wireless Personal Area Networks)

<sup>3</sup>(RF Radio frequency)

<sup>4</sup>(FH Frequency-Hopping)

<sup>5</sup>(WLAN Wireless Local Area Network)

<sup>6</sup>(AWMA Alternating Wireless Medium Access)

### 2.3. REDES MANET Y MESH

---

- Adaptive Packet Selection: técnica que permite seleccionar entre varios tipos de paquetes, por ejemplo el tamaño de la carga.
- Adaptive Packet Scheduling: técnica que realiza estimación de interferencia y políticas de programación maestra.
- Chanel Clasification: técnica que dispone de un algoritmo para determinar la calidad de los canales, utilizando métricas como RSSI, Carrier Sensing, y ACK<sup>7</sup> de los paquetes, puede ser medido utilizando técnicas de “Batch” o con técnicas online y offline.
- Adaptive Frequency Hopping: mecanismo que cambia de manera dinámica la frecuencia de los saltos para evitar interferencias. Es utilizado para la coexistencia entre dispositivos 802.15.1 que utilizan la banda 2 GHz como banda estática.

#### **IEEE 802.15.3 Especificaciones de Wireless Medium Access Control y Capa Física para WPAN**

El protocolo y la compatible interconexión de datos y comunicación multimedia equipada en transmisiones a 2.4 GHz en WPAN utilizando múltiples formatos de modulación para soportar velocidades escalables, se encuentra definido en este estándar. El protocolo subcapa Medium Access Control (MAC) soporta tipos de datos síncronos y asíncronos[2].

---

<sup>7</sup>(ACK acuse de recibo)

## Capítulo 3

# Redes de Sensores Inalámbricas

### 3.1. Definición

Una red de sensores inalámbrica es un sistema distribuido formado por nodos, también llamados motes, que interactúan con el entorno físico. Un mote es un pequeño dispositivo compuesto de un microprocesador con memoria, uno o varios sensores, una radio de baja potencia, una fuente de energía y un actuador. Cada nodo puede contar con diversos tipos de sensores, tales como térmicos, biológicos, químicos, ópticos o mecánicos, que miden las propiedades del ambiente. La información recabada por los sensores es generalmente procesada por los nodos. Sin embargo, debido a las limitaciones en memoria y capacidad de procesamiento, los nodos cuentan con un radio para transmitir de manera inalámbrica únicamente la información requerida y parcialmente procesada. Esta información es enviada hacia una estación base, como por ejemplo, una computadora portátil o una tableta electrónica. La estación base recibe la información de los nodos, la procesa si es necesario y lleva a cabo las acciones necesarias, como por ejemplo, informar a los usuarios si algún evento de interés ha ocurrido. Generalmente la fuente primaria de energía de los nodos son baterías, y en algunos casos del medio ambiente se obtiene una segunda fuente de energía (por ejemplo, utilizando paneles solares). Por otra parte, los actuadores pueden ser incorporados a los nodos dependiendo del tipo de sensores y de la aplicación.

La Fig. 3.1 muestra un ejemplo de una aplicación WSN de monitorización. En el ejemplo, la aplicación monitoriza el entorno físico de un viñedo, midiendo condiciones de humedad, temperatura, aparición de plagas, entre otras.

Las aplicaciones de las WSN puede clasificarse en dos categorías:

1. Monitorización

- a) Ambiente interior/exterior

### 3.2. ESTÁNDAR IEEE 802.15.4

---

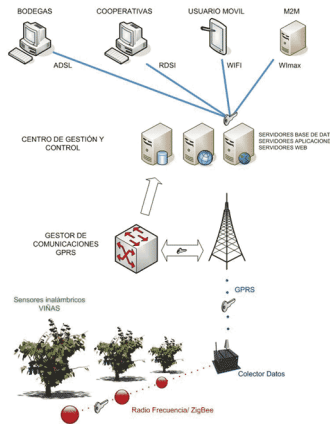


Figura 3.1: Ejemplo de WSN

- b) Salud y bienestar
- c) Energía
- d) Localización de inventario
- e) Fabrica y automatización de procesos
- f) Sismos
- g) Estructuras

#### 2. Rastreo

- a) Objetos
- b) Animales
- c) Humanos
- d) Vehículos

### 3.2. Estándar IEEE 802.15.4

Es un estándar propuesto para WPAN en bajas velocidades<sup>1</sup> [21]. El estándar está enfocado en el desarrollo a bajo costo. Está diseñado para aplicaciones de sensores inalámbricos que requieren un rango corto de comunicación para maximizar la vida de la batería. El estándar permite la formación de topología estrella y peer-peer entre dispositivos de la red. El estándar 802.15.4 está limitado a la capa física y capa MAC del modelo OSI.

El IEEE 802.15.4 investiga soluciones a velocidades bajas de datos con tiempo de vida de baterías desde multi-mes hasta multi-año. Opera en una banda

---

<sup>1</sup>(LR-WPAN low rate Wireless Personal Area Network)

### 3.2. ESTÁNDAR IEEE 802.15.4

---

de frecuencia internacional sin licencia. Aplicaciones potenciales son sensores, juguetes, insignias inteligentes, control remoto y automatización del hogar.

Las características del protocolo son [38]:

- Extremadamente bajo costo
- Facilidad de instalación
- Transferencia de datos segura
- Operación a corto alcance
- Razonable vida de la batería

Se clasifican dos dispositivos dentro del protocolo:

- Full function device (FFD)
  - Cualquier topología
  - Capacidad de coordinar una PAN
  - Se comunica con cualquier otro dispositivo
  - Implementa el protocolo completo
- Reduced function device (RFD)
  - Limitado a una topología de estrella (ver Fig. 3.2) o dispositivo final en una red peer to peer (ver Fig. 3.3).
  - No puede ser un coordinador de una PAN.
  - Implementación muy simple.
  - Protocolo reducido

Los dispositivos en la topología de estrella (ver Fig. 3.2) se comunican con el controlador central. Mientras que en la topología peer-peer (ver Fig. 3.3) ad hoc y autoconfigurables redes pueden ser formadas. Los dispositivos están diseñados para soportar los protocolos de la capa física y la capa de enlace. La capa de enlace soporta 868/915 MHz en banda baja y 2.4 GHz en banda alta. La capa MAC controla el acceso al canal de radio utilizando el mecanismo CSMA-CA. La capa MAC además es responsable de validar los frames, interface de la red, sincronización de la red, asociación de dispositivos y servicios seguros.

Opera con bajo consumo de energía. Para ello los dispositivos duermen para extender el periodo. Permiten el control del estado receptor por los niveles superiores. El servicio de administración permite el acceso a la PIB<sup>2</sup>, la asociación y la desasociación, notificación a nodos, exploración de la red, sincronización de la red, búsqueda.

El estándar IEEE 802.15.4 reduce las capas superiores del modelo OSI a solo dos (ver Fig. 3.4). La capa de red provee de la configuración de la red,

---

<sup>2</sup>(PIB PAN informativo base [6])

### Topología Estrella

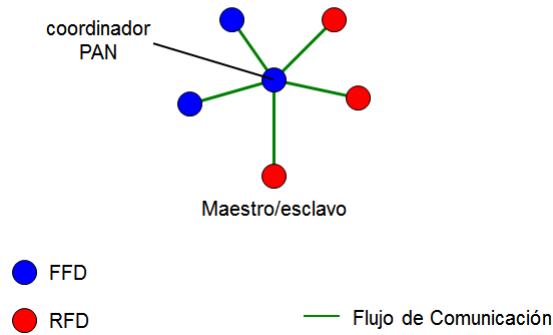


Figura 3.2: Topología de Estrella [38]

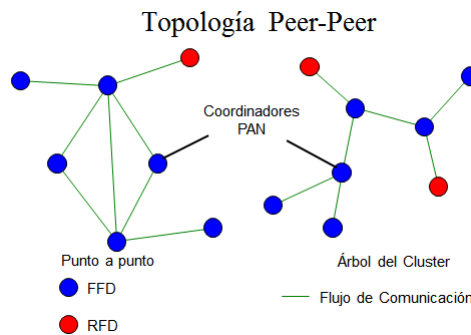


Figura 3.3: Topología Peer-Peer [38]

manipulación y encaminamiento de mensajes. La capa de aplicación provee las funciones de los dispositivos sobre la red en el estándar IEEE 802.15.4. La capa de red y la capa de aplicación no están completamente definidas en el estándar IEEE 802.15.4.

Los estándares desarrollados para las WSN tienen como requerimiento el bajo consumo de energía. Un estándar define la funcionalidad y protocolos necesarios para que los nodos sensores interfieran con la variedad de redes.

ZigBee [4] define los protocolos de la capa mas alta de comunicación en el IEEE 802.15.4 para LR-PANs. ZigBee es simple, con bajo costo y tecnología de comunicación inalámbrica de bajo consumo utilizado en aplicaciones embebidas. Los dispositivos ZigBee pueden formar redes malla conectándose miles de dispositivos entre ellos. Los dispositivos ZigBee utilizan poca energía y pueden operar con una batería de una celda por muchos años.

Existen tres tipos de dispositivos:

1. ZigBee coordinador: inicializa la formación de la red, almacena informa-



### 3.2. ESTÁNDAR IEEE 802.15.4

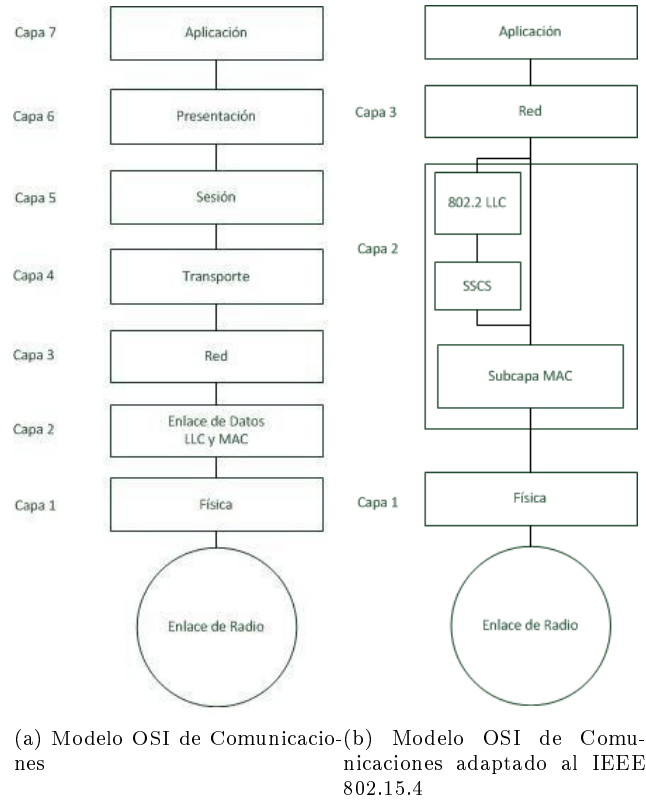


Figura 3.4: Adaptación de Modelo OSI en IEEE 802.15.4

ción y puede crear un puente entre redes.

2. ZigBee router: un grupo de dispositivos y provee comunicación multi saltos a través de los dispositivos.
3. ZigBee dispositivo final: consiste en sensores, actuadores, y controladores que recopilan datos y se comunican únicamente con el router o con el coordinador.

WirelessHART [17] protocolo de comunicaciones de red de malla inalámbrica para aplicaciones de automatización de procesos. El estándar es basado en IEEE 802.15.4 para funcionamiento en low power 2.4 GHz. WirelessHART es compatible con todos los dispositivos existentes, herramientas y sistemas. WirelessHART es simple, seguro y eficiente con la energía. Soporta red en malla, salto de canal y mensaje de tiempo sincronizado. La comunicación de la red es segura con estacionario, verificación, autenticación y administración de llave. La administración de la energía habilita a los dispositivos inalámbricos para mas

### 3.2. ESTÁNDAR IEEE 802.15.4

---

energía eficiente. WirelessHART está diseñado para soportar malla, estrella y topologías combinadas. Una red WirelessHART consiste de:

- Dispositivos inalámbricos de campo: los cuales son conectados a un proceso o equipo.
- Gateways: habilitan la comunicación entre los dispositivos inalámbricos de campo y las aplicaciones host.
- Controlador de automatización de procesos: sirve como un solo controlador para el proceso continuo.
- Host de aplicación.
- Administrador de red: configura la red y planifica la comunicación entre los dispositivos, además administra el encaminamiento y el tráfico de la red. Puede integrarse en el gateway, en el host de aplicación o en el controlador de automatización de procesos.

ISA100.11a [23] el estándar está diseñado para monitoreo inalámbrico con velocidad baja de datos y para aplicaciones de procesos de automatización. Define las especificaciones para la capa OSI, seguridad y administración del sistema. El estándar se enfoca en el bajo consumo de energía, escalabilidad, infraestructura, robustez e interoperabilidad con otros dispositivos inalámbricos. Las redes ISA100.11a utilizan únicamente el radio a 2.4GHz y salto de canal para incrementar la confiabilidad y minimizar interferencias. Puede funcionar en topologías de malla y estrella. Provee una simple, flexible y escalable funcionalidad de seguridad.

6LoWPAN<sup>3</sup> habilita la comunicación de paquetes IPv6 sobre redes basadas en IEEE802.15.4. Un dispositivo de bajo consumo puede comunicarse directamente con dispositivos IP utilizando el protocolo IP. Utilizando 6LoWPAN, los dispositivos de bajo consumo obtienen todos los beneficios de la comunicación IP y la administración. Debido a que los paquetes IPv6 tienen un tamaño mayor que el frame de IEEE 802.15.4 se utiliza una capa de adaptación. La capa de adaptación tiene toda la funcionalidad para la compresión de la cabecera. Con la compresión de la cabecera se tienen paquetes mas pequeños, creados al tamaño de un frame de IEEE 802.15.4. El mecanismo de administración de direcciones maneja la formación de las direcciones de comunicación de los dispositivos.

Wibree [41] es una tecnología de comunicación inalámbrica diseñada para consumo bajo de energía, comunicación a corta distancia y dispositivos de bajo costo. Wibree permite la comunicación entre pequeños dispositivos y dispositivos Bluetooth. En los dispositivos pequeños se encuentran relojes, teclados inalámbricos, sensores de deportes, que se conectan a dispositivos host como computadoras personales o celulares. Wibree funciona en 2.4 GHz y tiene una tasa de transmisión de 1 Mbps. El enlace de comunicación entre los dispositivos se encuentra entre 5 y 10 metros. Wibree está diseñado para trabajar con

---

<sup>3</sup>IPv6-based Low power Wireless Personal Area Networks

### 3.3. DESARROLLO A LA FECHA

---

Bluetooth. Bluetooth con Wibree hacen los dispositivos mas pequeños y mas eficientes.

### 3.3. Desarrollo a la fecha

Los estudios realizados para clasificar las técnicas de desarrollo siguen ciertas perspectivas. Las cuales están clasificadas en técnicas para áreas residenciales, técnicas para eventos en bosques y técnicas para la detección temprana.

En las técnicas para áreas residenciales se describen los estudios hechos a eventos que pueden llegar a ocurrir en lugares como edificios, lugares públicos, áreas donde se encuentran muchos civiles.

En las técnicas para eventos en bosques se describen los estudios a eventos que llegan a afectar los bosques como son incendios, los cuales generan una gran pérdida ambiental.

En las técnicas para detección temprana se mencionan métodos que se espera llegaran a detectar y de cierta forma intentar predecir los que puede llegar a ocurrir con anticipación.

#### 3.3.1. Técnicas para áreas residenciales

En los edificios de alto riesgo [57]. La arquitectura de la red esta comprendida por detectores, repetidores y un centro local en cada piso del edificio ver Fig. 3.5. Consiste en enviar alarmas de fuego automáticamente, creando una arquitectura de red y un protocolo de comunicación. El prototipo fue realizado con 11 detectores, 3 repetidores y un centro de vigilancia distribuidos en un piso del edificio. Los detectores responsables de la detección de fuego envían los datos a los repetidores los cuales envían el reporte de la información monitoreada. Los repetidores no únicamente monitorizan si no que permiten tener acceso a la red a los detectores. Los reportes de cada piso del edificio. La red puede estar comprendida por 255 repetidores y cada repetidor puede tener 255 detectores cada repetidor tiene una dirección de red asignada por el centro de vigilancia.

Debido a que fue diseñado se selecciono el RF Chipset CC1100 de Texas Instrument para los detectores y los repetidores. CC1100 es un transmisor de bajo costo diseñado para aplicaciones wireless de bajo consumo de energía. El formato de los frames enviados desde el transmisor contiene bits de preámbulo, cadena de sincronización, longitud, dirección, datos y CRC-16 (ver Cuadro 3.1). Debido a que el RF consume mucha energía únicamente se envían reportes ciertos periodos de tiempo.

bits preambulo (1010 ... 1010)	Palabra de sincronización	Campo Tamaño	Campo Dirección	Campo Datos	CRC- 16
8xn	16/32	8	8	8xn	16

Cuadro 3.1: Formato del paquete para CC1100 en bits[57]

### 3.3. DESARROLLO A LA FECHA

---

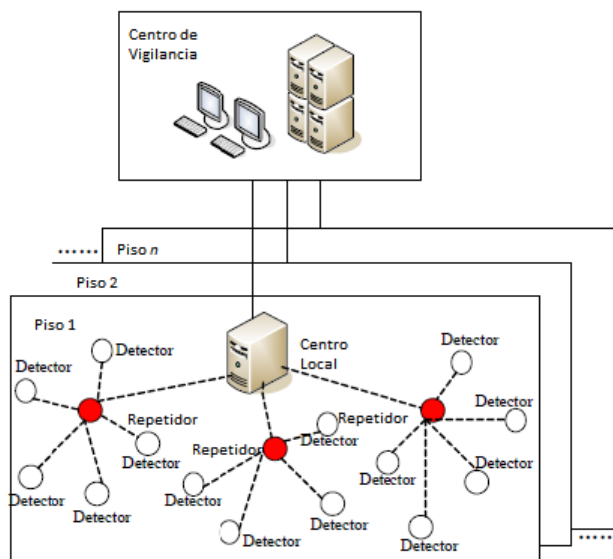


Figura 3.5: Arquitectura de la red [57]

Los repetidores reciben un mensaje de confirmación el cual registra la información de un nuevo nodo, guardándola en una tabla de vecinos, en donde se incluye el número serial, dirección, estatus, etc.

Los detectores reportan la información de detección a los repetidores una vez que se ha detectado una alarma de fuego. El formato que tiene el mensaje comprende el tipo de paquete, dirección fuente, dirección destino, contador, estatus de alarma, concentración de humo, nivel de batería. Cuando el repetidor recibe el mensaje del detector; el repetidor envía un mensaje al centro de vigilancia con el siguiente formato: tipo de paquete, dirección fuente, dirección destino, periodo de reporte, periodo wakeup, alarma y reset. Cuando el mensaje es recibido por el repetidor, este envía un mensaje de confirmación al detector. Para la detección se utiliza un sensor de humo photoelectronico y/o de temperatura.

Para áreas civiles como pueden ser tiendas departamentales, estaciones de tren, construcciones públicas. Se creó un framework [56] el cual se piensa soportar nodos defectuosos por ejemplo nodos que tienen corto alcance de comunicación. Es un framework para la detección de fuego y soporte de rescate. Sugiere 4 componentes en la implementación. Se realizaron experimentos para encontrar el intervalo de sensado idóneo de acuerdo con el número de sensores. Los eventos de carácter urgente son enviados lo más pronto posible. Los nodos se utilizan fuera por lo cual hacen difícil el manejo de la red directamente, por lo cual el framework permite un control remoto indirecto. El framework permite re-utilizar la información recabada para cierto propósito. El framework está compuesto por 4 componentes de operación:

1. WSN para la detección de fuego: la red está compuesta por nodos, los

### 3.3. DESARROLLO A LA FECHA

---

nodos pueden monitorear y recolectar datos de temperatura, iluminación, humo, etc.

2. Capa de recolección de información: el principal propósito es entregar la medición y los eventos de alarma de fuego al sink. Cuando el fuego ocurre la infraestructura puede ser destruida fácilmente por lo cual se puede apoyar con los nodos vecinos.
3. Middleware: Se encuentra en la Terminal (PC) conectada al sink. Los datos obtenidos son mostrados en la Terminal. El middleware puede enviar la información hacia otros programas como es el software de monitoreo.
4. Monitoring software: aquí se muestran los experimentos y demostraciones.

Para la implementación los nodos tienen un algoritmo de operación, reciben mensajes de otros nodos, pueden tomar decisiones, y pueden notificar. El algoritmo puede indicar las áreas que son seguras para escapar, las notificaciones se hacen al sink.

Para los experimentos se busco el IMR (interval-message-ratio). El IMR se calcula como se muestra en la ecuación (3.1):

$$IMR = \frac{\text{Intervalo de mensajes generados}}{\text{Intervalo de mensajes recogidos por el middleware}} \quad (3.1)$$

Para las pruebas se utilizo únicamente el valor de la temperatura. Con lo que el intervalo ideal para obtener los mensajes en el experimento es de 1250 y 1500 milisegundos, logrando evitar una sobresaturación de la red (ver Fig. 3.6).

Para interiores de edificios [5] se utiliza un monitoreo en el que los sensores se comunican con una estación central. La estación base esta conectada a una computadora, los motes se comunican a ella a través de la estación base. Los usuarios puede interactuar con la red de sensores de dos maneras:

1. Los usuarios remotos pueden acceder a una replica del servidor base en Berkeley
2. Se puede utilizar un pequeño dispositivos parecido a un PDA para interacciones locales.

Con la ayuda del servidor de base de datos más aplicaciones se pueden integrar para monitorear el sistema. Los datos leídos son enviados al servidor central vía red GSM. El servidor central esta conectado con una base de datos en línea. Desde esta base de datos una aplicación remota puede ser ejecutada vía Web y accesada desde una computadora, laptop o PDA. El modelo del sistema de monitoreo utiliza el protocolo ZigBee el cual esta diseñado específicamente para monitorización y control. La distancia para monitorización dentro de edificios se encuentra en el rango de 25 a 30 metros. La estación base continuamente se encuentra monitoreando a una frecuencia de 2.4 GHz cuando detecta un paquete detiene la monitorización y toma al paquete y lo envía a la computadora a

### 3.3. DESARROLLO A LA FECHA

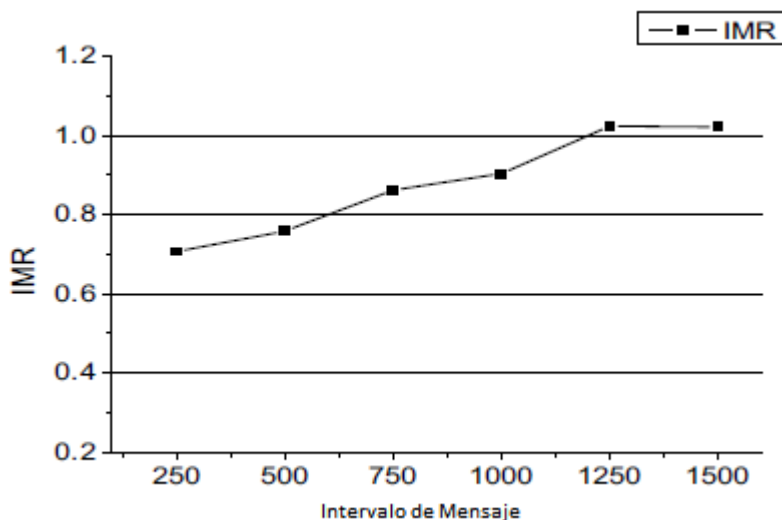


Figura 3.6: Resultado de los Experimentos [56]

57,6000 bps vía cable serial. La comunicación serial es controlada por UART (Universal Asynchronous Receiver/Transmitter). El servidor que hace las alertas esta escrito en lenguaje Java. La alerta consiste en un mensaje que informa el área donde se detecto fuego y el programa reproduce un archivo. Cuando el sensor se detiene de enviar paquetes durante un tiempo tolerable, el programa concluye una falla y dice que es necesaria una futura prueba. El intervalo asignado es de 1 minuto. El mensaje que llega a la estación base contiene el ID del nodo, un contador, grupo del mensaje, lectura de la luz, lectura de la temperatura (ver Cuadro 3.2). Cuando se recibe el mensaje se guarda con lo que se obtiene de del objeto `java.util.Date`. La base de datos lleva un historial de las lecturas grabadas. Los resultados obtenidos del sistema dan un eficiente monitoreo y habilitan la comunicación de los sensores con una estación central.

Dirección destino	ID del Manejador AM	ID de grupo	Tamaño del mensaje	ID de la fuente	Contador	Canal del sensor	Lecturas del sensor
2 bytes	1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	20 bytes

Cuadro 3.2: Formato del paquete[5]

En [32] para la detección en áreas residenciales sugiere la obtención de datos convenientes como son los de  $CO$  (Monóxido de Carbono),  $CO_2$  (Dióxido de Carbono), photocell, taguchi y temperatura. El aumento de concentración de  $CO$  no siempre indica fuego, mas sin embargo el aumento de concentración de  $CO_2$  si puede indicar flamas/fuego. En un estudio entre la combinación de humo

### 3.3. DESARROLLO A LA FECHA

---

y *CO* se concluye que los detectores *ION* dan una ventaja para la detección de llamas de fuego. Los photo detectores son beneficiosos para la detección de eventos que no involucren flamas de fuego. Una combinación entre *CO* e *ION* aumenta el acierto para la detección de fuego.

Los detectores de *ION* son más beneficiosos para la detección de flamas de fuego. Los detectores photoeléctricos no son beneficiosos para eventos de fuego. Los sensores de photoeléctricos e iónicos son muy sensibles al ruido. Los aumentos muy elevados ayudan mucho en la detección de fuego. Sensores de *CO*, *ION* y temperatura son más inmunes.

#### 3.3.2. Técnicas para eventos en bosques

Para la detección de fuego en el bosque se ha llegado a implementar redes neuronales para prolongar el tiempo de vida de las redes de sensores. En [30] los nodos se ubican en el bosque toman los datos y los envían a un cluster el cual lo envía al sink (ver Fig. 3.7). Para el funcionamiento de la red se puede reducir la sobre comunicación y el consumo de energía en las WSN. Madden [47] propuso que dentro de una red se puede manipular por un estilo SQL. Este aprovecha que es posible ejecutar consultas en los sensores y proveer oportunidades de optimización. La propuesta para la detección de fuego consiste en un diseño que detecta fuego en el bosque a tiempo y previniendo peligros de fuego en el bosque de manera más precisa.

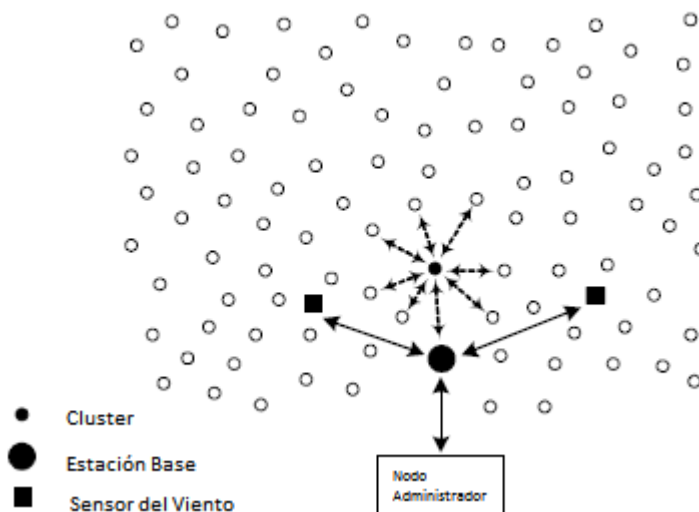


Figura 3.7: Una red de sensores inalámbrica para la detección de incendios forestales en tiempo real [30]

Los nodos toman los datos de la temperatura humedad y humo, algunos cuentan con GPS. El cluster se selecciona dinámicamente por un algoritmo

### 3.3. DESARROLLO A LA FECHA

para dar un balance en el consumo de la energía de todos los nodos. Para la colección de datos cuando los nodos encuentran algún evento anormal, entonces inmediatamente genera y envía un paquete ER (Emergence Report) con la información del evento anormal. Para el procesamiento de los datos cuando el sink recibe un ER lo considera de alta prioridad, si recibe un mensaje QR(Query Response) el cluster aplica un algoritmo de agregación. Para el procesamiento de paquetes RR (Regular Report) se construye un índice de clima, en el índice se encapsula en un PR (Processed Report), el PR se envía por el cluster al sink (ver Fig. 3.8).

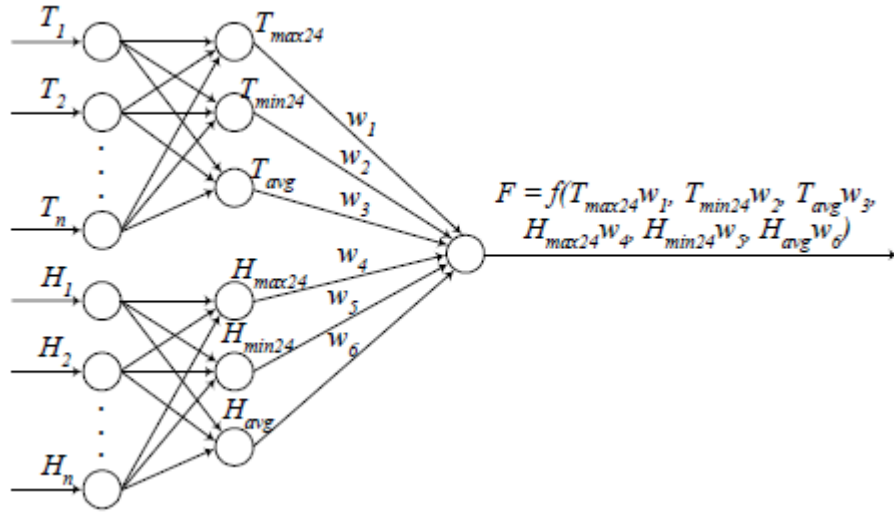


Figura 3.8: Procesamiento de Paquete RR usando el método de red neuronal [30]

Se utilizan redes neuronales para prolongar el tiempo de vida de las redes de sensores. Se comenta que los sensores pueden ser utilizados para ayudar en el pronóstico de fuego en bosques apoyándose con imágenes satélites.

El sistema [32] de Monitoreo FWI toma en consideración la temperatura, la humedad relativa, el viento y lluvia. En base a estos valores se genera un valor el cual se compara con el código FFMC Ignition Potencial Based. El sistema FWI cuenta con buenos avances para monitoreo en bosques. La estructura del Sistema FWI se observa en el cuadro 3.9.

Para la detección de fuego [44] en bosques se proponen dos métodos:

1. Threshold Method
2. Dempster-Shafer Method

**Threshold Method** (Método del Umbral) método con el que se proponen 7 estados empezando desde cero, toma en base los datos de la temperatura y la humedad. **Dempster-Shafer Method** método basado en hipótesis, cuando



### 3.3. DESARROLLO A LA FECHA

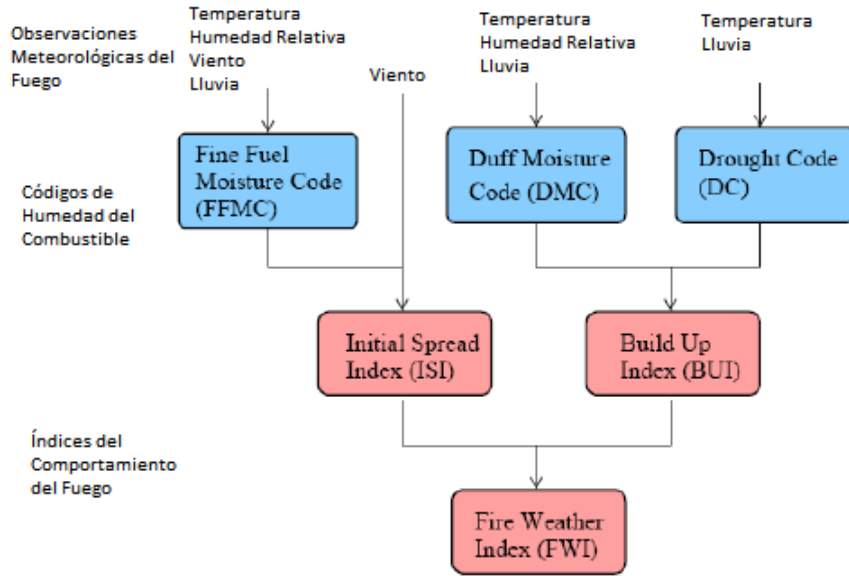


Figura 3.9: Estructura del Sistema FWI [32]

existe fuego y cuando no existe fuego. Para la adquisición de datos en los dos métodos se realizaron experimentos con los motes y fuego artificial (antorchas) con una duración del fuego de 5 a 15 minutos y distancias de 50 a 100 centímetros. En los experimentos lo más importante a destacar es ver los valores de falsos negativos los cuales se consideran con mayor relevancia, que los falsos positivos.

Los dos métodos probados mostraron resultados muy diferentes en los que se encuentra que el método Threshold tiene menos falsos negativos en los eventos. El porcentaje de aciertos es mayor en el método Threshold. Se recalca en el artículo que aunque el método Threshold resulto con mayores resultados en cuanto a aciertos, el método Dempster-Shafer muestra una ventaja en permitir una extensión rápida de nuevas evidencias, cargándose con mayor eficiencia. Se busca en un futuro lograr distinguir la luz de los rayos del sol con la luz del fuego, ya que por esa razón mostró en las pruebas falsos negativos. La máquina de estados utilizada con Threshold Method es la que se muestra en la Fig. 3.10.

Para la monitorización en el bosque [12] se realizó un experimento en campo donde se construyeron bases donde se montaron los motes (MTS420), Mica2 con GPS. Se realizó un experimento donde se colocaron 10 motes, donde 6 se encontraron dentro del área de fuego y 4 próximos al fuego. La arquitectura de del sistema se muestra en la Fig. 3.11. Cuando se tomaron los datos se considera los valores de temperatura, humedad, presión y posición geográfica. Los datos son enviados a la estación base para enviarlos a un servidor base de datos (MySQL).

### 3.3. DESARROLLO A LA FECHA

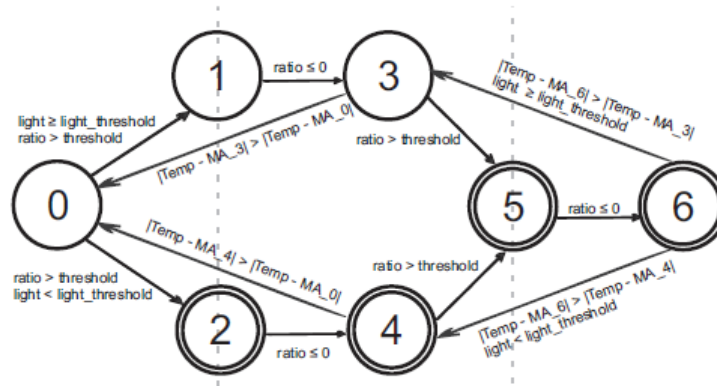


Figura 3.10: Máquina de Estados para definir las condiciones del ambiente [44]

Se realizaron dos experimentos en los cuales se llegaron a destruir algunos motes, algunos dejaron de enviar datos debido a que se cortó la alimentación (Batería). Cuando ocurrió el evento de fuego se observa que la humedad baja y la temperatura aumenta, la presión atmosférica disminuye. Como observación se ve que se requieren 4 minutos para que la presión barométrica regrese a su valor inicial.

En [26] se propone la arquitectura para una WSN llamada FireNet para ayudar a los bomberos. La arquitectura de FireNet (ver Fig. 3.12) consiste en:

- Departamento de bomberos.
- Autos de rescate con laptop donde se encuentra el sink.
- Cada bombero posee un mote con sus datos (nombre, edad, especialidad, temperatura, humedad, entre otros datos).

El software primero necesita analizar la información recolectada del fuego para detectar eventos, después estos datos son enviados al comandante de incidentes y al departamento de bomberos. Después el software genera automáticamente la planificación del proceso de rescate basándose en los datos recolectados, por último el software notifica al comandante de incidentes cuando eventos importantes son detectados. Cada bombero lleva un sensor el cual toma las mediciones de la temperatura, humedad, químicos.

En [13] se describe un componente llamado Local Alerting Control Unit (LACU), así como un método para la estimación de la localización del fuego. La arquitectura de LACU (ver Fig. 3.13) esta dividida en cinco componentes:

1. Proxy de comunicación
2. Proxy Sensor
3. Componente de Fusión

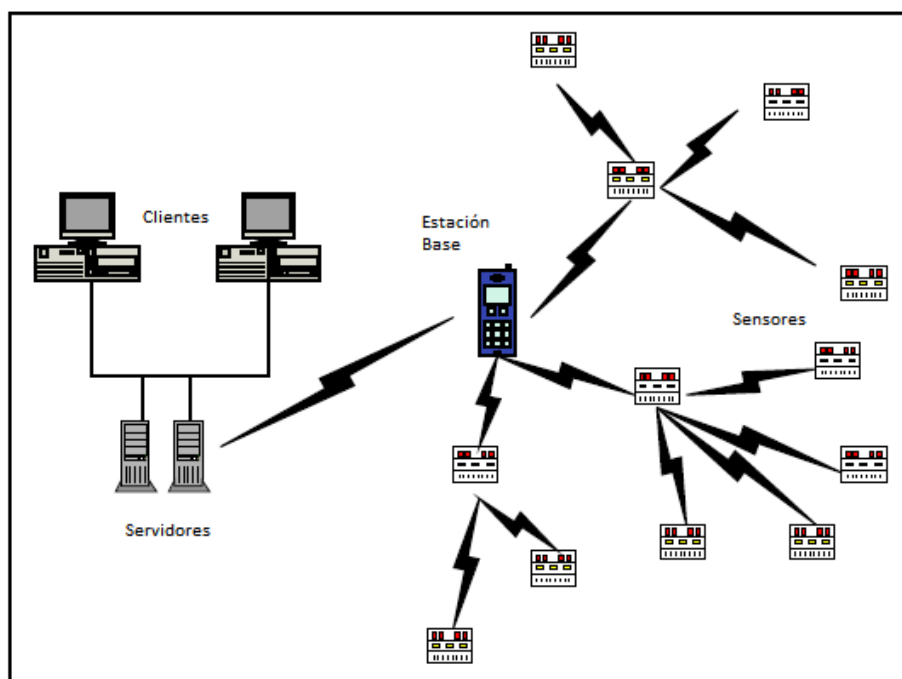


Figura 3.11: Arquitectura del sistema de monitoreo [12]

#### 4. Componente de Alerta

#### 5. Base de Datos

El Proxy de comunicación es el responsable de la comunicación entre LACU y el subsistema de cómputo. El Proxy sensor: recibe y almacena las lecturas transmitidas por los sensores distribuidos en el área a monitorear. El componente Fusión evalúa los datos medidos y determina si llega a ocurrir fuego. El componente de Alerta es activado por el componente de Fusión, provee notificaciones al subsistema de cómputo y a los usuarios cuando una situación de emergencia ocurre. El componente Base de Datos almacena el histórico, de la identificación de cada sensor y la localización que se provee por el componente Proxy sensor.

Los sensores son ubicados en dos áreas:

- Public LACU: instalados y operados por autoridades públicas.
- Private LACU: son instalados por los ciudadanos para proteger sus propiedades privadas.

Para la detección de fuego se probó con una red de sensores, tomando en cuenta las lecturas de la temperatura muestra buenos resultados, en ocasiones suele dar



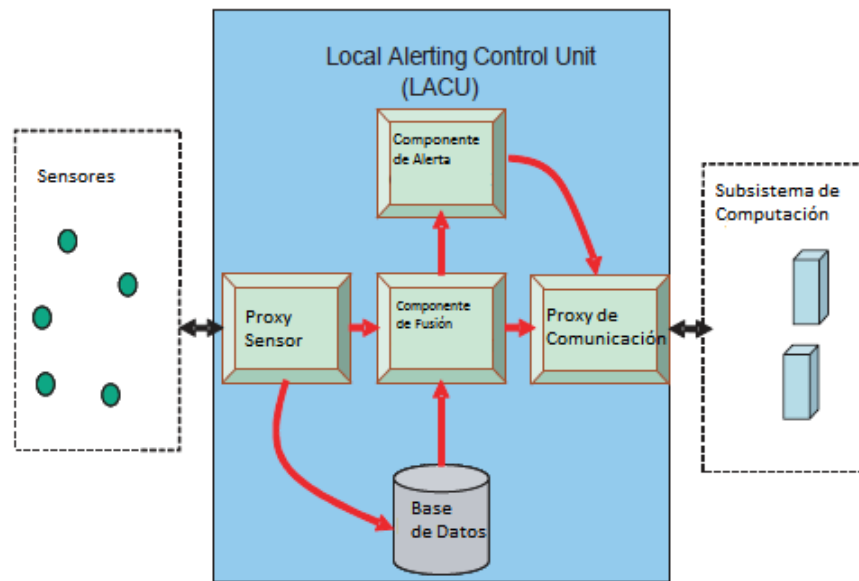


Figura 3.13: Arquitectura de LACU [13]

- **Componente de Fusión:** es donde se evalúan los datos de los sensores y se determina si existe fuego en el área. Además estima la localización exacta del fuego.
- **Componente de Alerta:** este es lanzado por el Componente de Fusión y provee notificaciones a CS cuando la situación de emergencia ocurre.
- **Base de Datos:** se encuentra el histórico de los datos, la identificación y localización de cada nodo, los datos son obtenidos del Proxy Sensor.

En gran parte CS esta basado en infraestructura computacional GIS y Grid donde se fusiona la información almacenada de los sensores, se procesa y visualiza usando datos adicionales del clima (velocidad del viento, dirección). Las principales funciones de CS son:

- Coleccionar y almacenar las mediciones de los sensores del área de interés.
- Realiza algoritmos de fusión de datos para evaluar el nivel de riesgo.
- Lanza una simulación en caso de alarma, mostrando lo que se estima ocurra en los siguientes 180 minutos.

La fusión de datos se lleva a cabo considerando el caso donde la temperatura en ambiente normal aumenta, de una manera anormal. La fusión de datos se lleva en dos niveles:

- El primero se realiza en LACU

### 3.3. DESARROLLO A LA FECHA

---

-El segundo en CS

En el primer nivel se toma encuentra la media de las temperaturas dependiendo de la hora, fecha y mes del año. Con el cual se estima de datos estadísticos, modelos empíricos, previsión o incluso de los mismos sensores. En el caso del fuego la humedad relativa disminuye al contrario de la temperatura que incrementa. Cuando se detecta riesgo de fuego en la primera capa, la segunda capa realiza el proceso de fusión con datos del sensor de visión (cámara) y los datos que le provee LACU.

Los nodos son ubicados en un área, toman los datos de la humedad y temperatura y son enviados al Proxy sensor que se encuentra en LACU. El mecanismo propuesto esta basado en fusión de datos multinivel. Los datos obtenidos del ambiente son la temperatura y la humedad. Para acoplar los diferentes tipos de sensores y entregar alarmas con gran acierto y confianza se adapto un esquema de fusión en capas. Diferentes sensores son procesados en un esquema de fusión en dos capas para mejorar la fiabilidad en el sistema de detección de fuego.

En [53] se describe la experiencia de la utilización de Firementor la cual se clasifica como una aplicación de monitoreo y logísticas de operación para la administración de incendios forestales (ver Fig. 3.14).

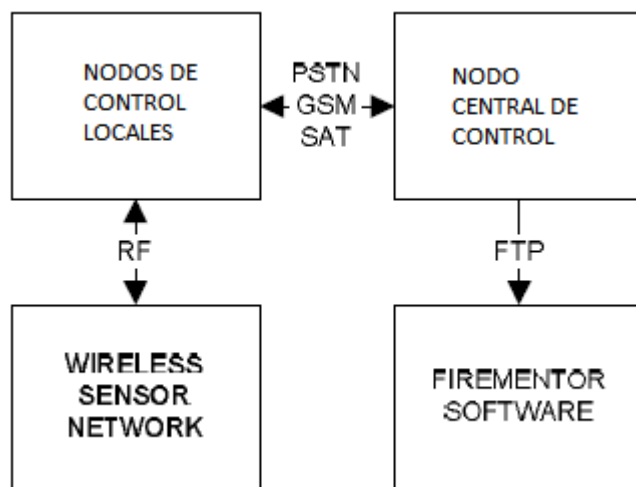


Figura 3.14: Estructura del sistema [53]

La WSN se comunica con CCN vía LCN. La comunicación entre CCN y LCN es GSM o PSTN. La comunicación entre CCN y Firementor es vía FTP. Cada segmento de WSN debe tener a los menos 2 LCN para comunicarse con CCN utilizando alternativos enlaces. Los LCN son responsables de enviar el estado de la WSN a CCN.

Fueron colocados 68 nodos y 2 LCN en el bosque de Nea Penteli. Los cuales utilizan TinyOS. Los mensajes que se envían incluyen:

- node id

### 3.3. DESARROLLO A LA FECHA

---

- node temperature
- voltage of battery

En las primeras pruebas se enviaron los mensajes cada 30 segundos con lo que la batería llega a durar aproximadamente 7 días. Si se cambia a modo normal los mensajes se envían cada 3 o 4 horas con lo que la duración de la batería es de 5 a 10 años.

Se construyeron 2 estructuras para alojar los sensores en uno se dejó la antena dentro y en el otro se dejó fuera. Dejando la antena fuera se observó que existe un aumento entre el 10-20 % en cuanto a la operación de en términos de pérdida de paquetes y distancia de operación.

Dejar más de un sensor con comunicación directa con el LCN aumenta el porcentaje de captura de mensajes. Específicamente se probó que si 3 nodos están en comunicación directa con el LCN la captura de mensajes incrementa de un 70 % a 95 %. Se realizó la prueba por 2 horas.

En [25] se propone una estructura para WSN para la detección de fuego forestal basándose en la técnica de ZigBee. La estructura de la red consiste en:

- WSN
  - nodo
  - cabeza del cluster
  - coordinador de la red
- Gateway
- Internet/red satelital
- Computadora de monitoreo

Los nodos son los responsables de tomar los datos de la temperatura y la humedad. Los datos son enviados al nodo cabeza del cluster. El nodo cabeza del cluster recibe los datos de los nodos les aplica el método de fusión de datos y transmite el paquete al coordinador de la red mas cercano. El coordinador de la red posee las básicas funciones de administración de red, como son configuración de la red, registro de equipo y control de acceso. La información la envía por comunicación inalámbrica. Cuando los gateway reciben los datos los almacenan en una base de datos local después transmiten los datos a la computadora que monitorea, aquí se toman las decisiones para la prevención de fuego.

En la Fig. 3.15 se muestra el diagrama de flujo para la transmisión de datos el cual inicia en la computadora de monitoreo. Monitorizar la temperatura y la humedad en los bosques se logra de una manera mas precisa, asegurando la transmisión de datos, construyendo una red flexible, con bajo costo y bajo consumo de energía.

En [9] se diseñó un sistema llamado FFSS (Forest-Fires Surveillance System) el cual consiste en una WSN, middleware y aplicación web (ver Fig. 3.16). La estructura del sistema es:

### 3.3. DESARROLLO A LA FECHA

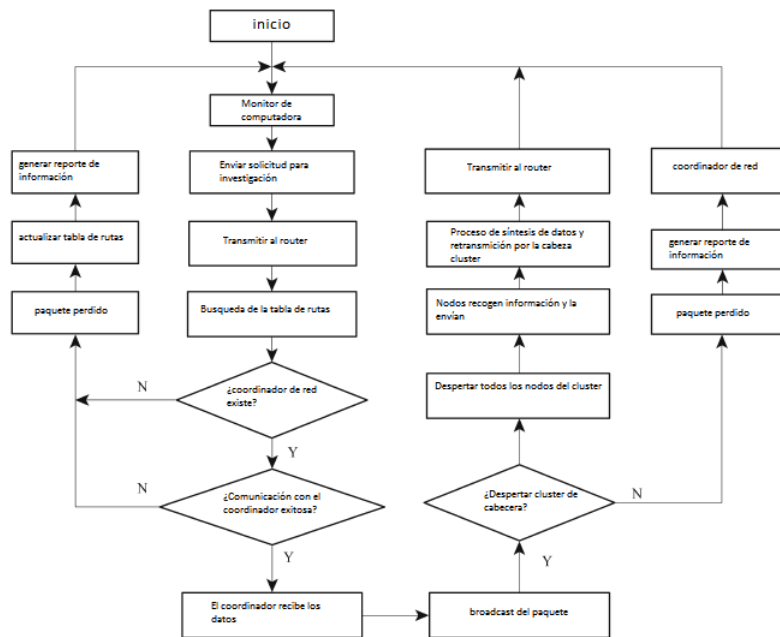


Figura 3.15: Transmisión de datos [25]

- WSN
- Sink
- FFSS
  - Base de datos
  - middleware
  - servidor web

Los nodos sensores funcionan con el sistema operativo TinyOS [3], toman los valores de la temperatura, la humedad y la luz. La red funciona con un protocolo diseñado para 3 fines: optimización, simplicidad y escalabilidad. El sistema FFSS determina el estado y nivel de riesgo de un fuego forestal con la fórmula de la Oficina Forestal. En el sistema se muestra los estatus recibidos que el administrador de monitoreo puede utilizar fácilmente. En la parte de middleware es donde se calcula el nivel de riesgo. En tiempo real se puede mostrar las gráficas de los datos recibidos de la temperatura, humedad y luz. La aplicación web consiste en base de datos y web-application, que es para que el administrador puede monitorear vía Internet.

La fórmula utilizada necesita cuatro días de muestra como mínimo para optimo funcionamiento, se basa en tomar en cuenta la humedad relativa. Se



### 3.3. DESARROLLO A LA FECHA

---

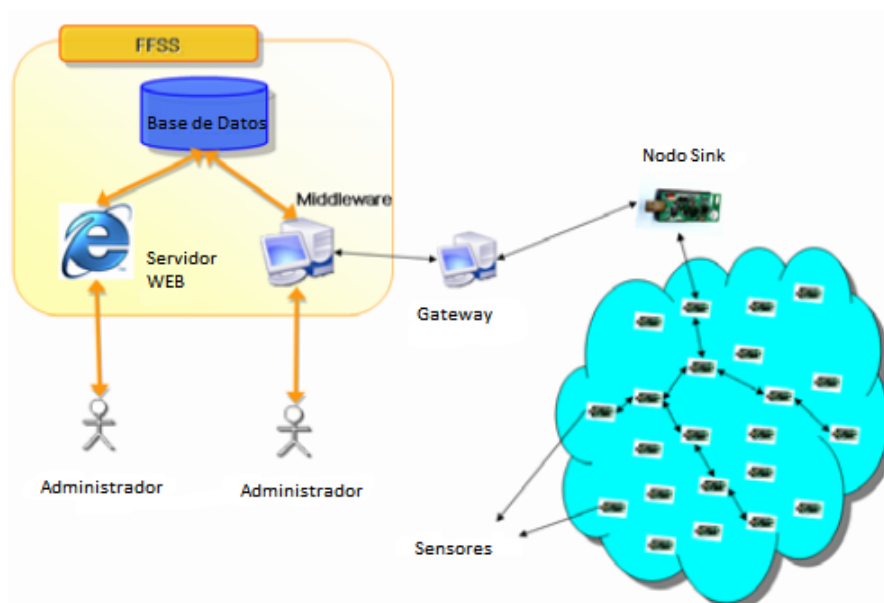


Figura 3.16: Estructura de los componentes de FFSS [9]

obtiene con la fórmula un valor, el cual se compara con una tabla de índices, el índice obtenido se clasifica en tres estados: extremo, alto y bajo. Las alarmas que genera el sistema FFSS a la estación de bomberos o a residentes vecinos, provee una rápida extinción de los fuegos forestales por lo tanto los daños y lesiones se han visto reducidos.

En [28] se propone un algoritmo para continuous skyline computation para recolectar datos de WSN para monitorear el ambiente de potenciales fuegos forestales. El algoritmo está compuesto por métodos los cuales son:

- método de nueva lectura
- método de agregar lectura al padre (parent)
- método para verificar skyline antigua.
- método agregar lectura

La sincronización entre nodos no es necesaria, porque cada nodo tiene su propio reloj, y en cada nodo se le aplica un tiempo para obtener el skyline. El skyline es la lectura más alta. En el método de nueva lectura primero se verifica si expiró la más antigua skyline. Después se verifica la lectura con las skylines presentadas, para ver si es dominada la lectura. Si es dominada se agrega a las lecturas dominadas por el skyline encontrado. En caso contrario se remueve del índice cada skyline dominado por la lectura y se propaga al siguiente nodo. En el método de agregar al padre (parent) se agregan las lecturas que son dominadas

### 3.3. DESARROLLO A LA FECHA

por otra lectura, muy parecido al anterior. En el método para verificar skyline antigua, si la skyline expiró se elimina y se agregan todas las hijo. En el método para agregar una lectura se verifican las skylines si domina la lectura, después se verifica que sea más reciente y se agrega una lectura hijo, si no se agrega la lectura a las skyline, después se propaga a los demás nodos.

Se realizó una simulación con diferente cardinalidad, utilizando diferentes tamaños en la ventana, la primer con 5000 y la segunda con un 10 por ciento de la cardinalidad. La cardinalidad va de 10 miles a un millón. Las simulaciones de realizaron en tres tipos de distribución de datos:

- independiente
- correlacionado
- anti-correlacionado

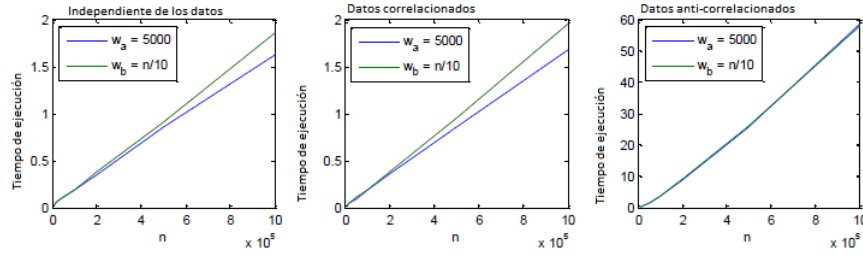


Figura 3.17: Resultados de los experimentos [28]

La simulación para independiente y correlacionado es muy similar sin variación notable, se observa que para grandes cardinalidades existe un incremento leve en el runtime de la ventana con el diez por ciento de la cardinalidad (ver Fig. 3.17). Esto es esperado debido a que los skylines duran más tiempo en el caso de ventanas más grandes. Para el anti-correlacionado se observó que el runtime es mayor considerablemente cuando se aumenta la magnitud de la ventana. Es esperado por que muchas lecturas son skylines en los datos anti-correlacionados.

#### 3.3.3. Técnicas para detección temprana

En [55] se propone el algoritmo VMBA para la forma de operar de la red. Cuenta con cuatro módulos los cuales son:

1. Modulo de escucha y filtración
2. Modulo de pruebas de contenido
3. Modulo de decisión de control de confianza
4. Modulo verifica los vecinos

### 3.3. DESARROLLO A LA FECHA

El primer modulo es responsable de examinar la validez de las mediciones recibidas de los nodos vecinos, filtrando las lecturas bajo un rango, con lo que se construye una tabla. El segundo modulo prueba el contenido de las tablas creadas en el primer modulo, se realiza para evaluar los datos, respecto a los asignados dinámicamente o estáticamente calculados de la norma estimada de el valor de la operación del nodo vecino. En el tercer modulo se concentra en el seguimiento de cambios de los nodos vecinos asignando un tiempo de ventana. En el cuarto modulo recibe y envía peticiones, con lo que verifica los nodos vecinos con un aviso de cambio en memoria, con lo que se reportan los nodos vecinos en el tiempo de monitoreo.

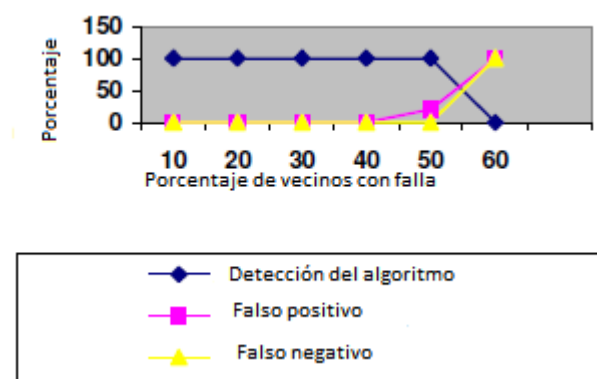


Figura 3.18: Mensajes de aviso con diferente numero de nodos vecinos defectuosos [55]

Para probar el algoritmo se modificaron dos componentes de TinyOS los cuales son MultiHopEngineM y MultiHopLEPSM. La técnica utilizada esta basada en módulos donde el primer modulo es el de mayor importancia debido a que es donde se generan las tablas que serán de utilidad para después tomar decisiones. El algoritmo mostrado detecta el 100 % de los nodos defectuosos cuando el porcentaje de nodos defectuosos es menor al 50 % (ver Fig. 3.18).

En [50] se propone un algoritmo para la detección de fuego tomando los valores de la temperatura, CO y densidad del humo, utilizando tecnología de fusión de datos. Esta compuesto por tres capas (ver Fig. 3.19):

1. Capa de Señales
2. Capa de características
3. Capa de decisión

La primera capa toma y procesa los datos. La segunda capa toma la característica del fuego de los datos de la primera capa, almacenándolos en una base de datos. La tercera capa fusiona las diferentes características de fuego obtenidas de la segunda capa y toma una decisión de la probabilidad de fuego. Los nodos son

### 3.3. DESARROLLO A LA FECHA

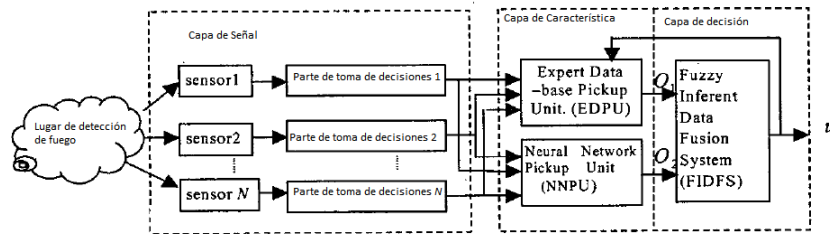


Figura 3.19: Estructura de las tres capas del sistema para la detección de fuego [50]

simulados en MATLAB, se combina con datos de la estandarización de llamas de fuego en China llamado SH4. El algoritmo para la fusión de datos de fuego se basa en una red neuronal la cual se encuentra en la segunda capa. Después de hacer las simulaciones utilizando las 3 capas se concluye que es posible predecir el fuego en un ambiente de llamas de fuego o algodón ardiendo.

En [54] se utiliza el framework Bayesian para resolver el problema cuando se tiene una secuencia de observaciones. Se pueden considerar en dos fases:

1. Primero los sensores toman una serie de observaciones
2. La serie de observaciones es enviada a Fusion Center

Lo primero que se realiza es tomar observaciones en cada sensor ya que se tiene una serie, esta se envía a Fusion Center el cual tomará las siguientes decisiones para los siguientes sensores. Los sensores cada cierto tiempo envían las observaciones tomadas y toda la decisión la tomara el Fusion Center.

Los experimentos se realizaron en tres casos:

1. se utilizo únicamente un sensor con el que se aplico la Función Local de Decisión (LDF) en los resultados se observo una distribución Gaussiana
2. se utilizaron dos sensores idénticos, lo diferente al primer caso es que aquí se utilizan dos valores de umbral en el LDF, los valores obtenidos son muy similares al primer caso.
3. se utilizaron dos sensores diferentes, se cambiaron las funciones.

El modelo matemático trata de solucionar dos problemas:

1. optimización de infinite-horizon
2. Minimizar finite-horizon

Los resultados obtenidos en el experimento para 1 y 2 sensores idénticos fueron muy similares mostrando que el infinite-horizon describe una función Gaussiana (ver Fig. 3.20 y Fig. 3.21).

### 3.3. DESARROLLO A LA FECHA

---

En [46] se proponen dos métodos para la detección de eventos. Los nodos fueron utilizados en forma simulada. Las simulaciones se llevaron a cabo para la probabilidad de detección y la probabilidad de falsa alarma.

Los modelos utilizados fueron:

- Para la detección de eventos con doble ventana deslizante.
- Detección híbrida basa en un sistema de lógica difusa (ver Fig. 3.22).

Se obtuvo en la simulación del método de doble ventana deslizante el 91.499 % para la detección y 0.08 % para falsa alarma. En la simulación del sistema de lógica difusa se obtuvo un 99.97 % de probabilidad de detección y un 0.05 % para falsa alarma.

En [48] se menciona la importancia de utilizar WSN para incendios forestales en Turquía. Se menciona la configuración (ver Fig. 3.23 en la página 42) básica de una WSN:

- Nodo sensor(SN)
- Cabeza del cluster(CH)
- Estación base(BS)

Los nodos sensores tienen básicamente 4 funciones:

- medir (sensar)
- procesar
- transmitir
- alimentación

En el nodo puede existir una aplicación que active una alerta cuando por ejemplo exista un exceso en el valor de la temperatura, enviando un reporte al CH. En el CH se puede realizar la agregación y procesamiento antes de enviar los datos a la BS. Los incendios forestales tienen en muchas dimensiones efectos negativos económica, social y ecológicamente. Entre mas datos a cerca de incendios forestales se logra una mejor administración del fuego. Las WSN son la decisión correcta y con menos costo para sistemas de detección temprana.

En [19] se muestra el proyecto llamado PAIS busca reducir los daños de los fuegos forestales con detección temprana y eliminación. La estructura que tiene PAIS es:

- CORs (Centro Operativo Regionale) coordina todas la actividades en escala regional
- COLs (Centro Operativo Locale) coordina todas las actividades en escala local
- UPRs (Unita Periferica di Rilevamento) unidades sensoras distribuidas en un territorio de acuerdo a evaluaciones de riesgo de fuego.
- Servicios de Fuego y rescate listos para intervenir en una situación de emergencia.

### 3.4. HERRAMIENTAS DE DESARROLLO

---

Cada COL administra un variable numero de UPRs, de 1 a 12, para operar el ambiente de monitoreo. Muchas COLs son ligadas a una COR para un cambio de datos e imágenes. Toda la información y alarmas vienen de un territorio regional, la COR eficientemente coordina a los bomberos. La UPR es montada en un vehículo todo terreno con GPS. La detección temprana de la vigilancia es realizada por SMWIR<sup>5</sup> y cámaras de espectro visible. Cada UPR puede controlar el área circundante con un dedicado sistema de monitoreo. Los sensores SMWIR funcionan con una temperatura constante menor a -45 grados centígrados. Por tal motivo se instalo un enfriador Peltier que cuenta con un controlador PID<sup>6</sup> para minimizar el error de la temperatura durante transitorios y constantes estados.

Se define la determinación para seis parámetros que garanticen las condiciones de control óptimas del sensor de temperatura. Se toman los valores de una cámara que detecta el espectro visible. La comparación entre el modelo basado en tuning de controladores PID y la obtenida con la propuesta HGA. Cuando se evaluó el rendimiento de HGA<sup>7</sup> tuning, la temperatura mostró un comportamiento senoidal (ver Fig. 3.24 en la página 43). Las pruebas fueron realizadas en condiciones ambientales normales con una visibilidad de 25 Km. El fuego utilizado en las pruebas lo podía observar el UPR sin obstáculos. En estas condiciones, el sensor IR es capas de detectar superficie de fuego de 5 metros cuadrados localizados a una distancia de 10 Km. En distancias cortas el sensor es capas de detectar superficies de fuego menores.

### 3.4. Herramientas de Desarrollo

Para desarrollar aplicaciones que optimicen el uso de los recursos de las WSN es necesario contar con una plataforma de software. Entre los componentes de la plataforma de software se encuentra el sistema operativo y el lenguaje de programación (compiladores, entornos de desarrollo, bibliotecas, etc.). Un sistema operativo es un componente de software que administra los recursos de hardware de un sistema informático y proporciona una interfaz a los programas de aplicación de los usuarios [51].

El diseño de un sistema operativo para redes de sensores es diferente al de un sistema operativo tradicional debido a las características específicas de las WSN, tales como recursos limitados, gran dinamismo y a la dificultad de acceso a los entornos en los que se distribuyen los nodos. A la fecha se han propuesto algunos sistemas operativos para redes de sensores [33].

A continuación se presentan algunos de ellos:

- *Bertha*. Una plataforma de software diseñada e implementada para modelar, evaluar e implementar una red de sensores distribuida de muchos nodos idénticos.

---

<sup>5</sup>Short and Medium Wavelength InfraRed

<sup>6</sup>Proportional-Integral-Derivative

<sup>7</sup>Hardware-Based Genetic Algorithm

### 3.4. HERRAMIENTAS DE DESARROLLO

---

- *Nut/OS*. Es un pequeño sistema operativo para aplicaciones en tiempo real, que trabaja con CPUs de 8 bits.
- *Contiki*. Es un Sistema Operativo de libre distribución para usar en un limitado tipo de computadoras, desde los 8 bits a sistemas embebidos en micro-controladores, incluidos *motes* de redes inalámbricas.
- *TinyOS*. Primer sistema operativo desarrollado específicamente para redes de sensores.
- *MANTIS*. Sistema Operativo escrito en C de código abierto basado en POSIX.
- *LiteOS*. Sistema operativo desarrollado en principio para calculadoras, pero que ha sido también utilizado para redes de sensores.

En [16], se presentan el estado del arte de los sistemas operativos para redes de sensores, con la finalidad de apoyar a diseñadores y programadores de aplicaciones para WSN en la elección del sistema operativo más adecuado al hardware y tipo de aplicación específica. Es importante considerar cuatro requerimientos de las aplicaciones para redes de sensores al seleccionar el sistema operativo y lenguaje de programación. Estos requerimientos son [52]:

- Uso eficiente de la energía
- Escalabilidad
- Resistencia a fallos
- Colaboración

El uso eficiente de la energía de los nodos es un requerimiento de todas las aplicaciones para redes de sensores. Debido a que la comunicación inalámbrica es la actividad que más consume energía, es importante reducir en la medida de lo posible el envío innecesario de información. Por esta razón, el modelo de programación utilizado debe ayudar a los programadores en el fácil desarrollo de aplicaciones con bajo consumo de energía.

La escalabilidad es un requerimiento importante debido a que muchas aplicaciones utilizan cientos de nodos. Uno de los factores que más influye en la escalabilidad es el limitado ancho de banda del estándar IEEE 802.15.4 y que es utilizado por muchos sistemas operativos para WSN. Debido a esta limitante, la principal preocupación relacionada con la escalabilidad en redes de sensores consiste en reducir la comunicación para utilizar eficientemente el limitado ancho de banda disponible. El modelo de programación utilizado debe permitir el desarrollo de aplicaciones escalables y mecanismos en tiempo de ejecución para lograr un uso eficiente del ancho de banda.

La resistencia a los fallos es también importante en aplicaciones que se ejecutan por un largo periodo de tiempo. Estas aplicaciones deben mantenerse funcionando a pesar de la existencia de un canal de comunicación poco confiable,

### 3.5. PLATAFORMAS

---

fallos en los nodos o fallos inesperados. El desarrollo de aplicaciones resistentes a los fallos y que sean capaces de adaptarse a nuevas condiciones de operación requieren del soporte del modelo de programación.

Finalmente, la colaboración es también un requerimiento importante, debido a que es posible utilizar de manera eficiente los recursos existentes (por ejemplo, el ancho de banda) si los nodos colaboran entre si al procesar y enviar la información. Sin embargo, el desarrollo de algoritmos colaborativos no es sencillo y se benefician en gran medida si pueden apoyarse en un buen modelo de programación.

Para cumplir con los requerimientos de las aplicaciones para redes de sensores es importante seleccionar, además del sistema operativo, un modelo de programación adecuado. A la fecha se han propuesto diversos modelos de programación para WSNs. Sugihara y Gupta en [52] clasifican a los modelos de programación en dos categorías: modelos de programación de bajo nivel y modelos de programación de alto nivel. Los primeros están centrados en la plataforma, se enfocan en la abstracción del hardware y permiten un flexible control de los nodos. Un ejemplo de este modelo es el lenguaje nesC [18] utilizado en conjunción con el sistema operativo TinyOS [20], que se han constituido en el estándar de facto en el desarrollo de aplicaciones para WSNs.

Los modelos de programación de alto nivel están centrados en la aplicación y tienen como objetivo facilitar el desarrollo de las aplicaciones, en contraste con el modelo anterior que busca optimizar el desempeño del sistema. Ejemplos de macro-lenguajes de esta categoría son TinyDB [31] y Regiment [40].

### 3.5. Plataformas

MICAz [35] funciona en el estándar IEEE 802.15.4 a 2.4GHz, diseñado específicamente para redes de sensores. Cuenta con una velocidad de 250 kbps en comunicación radio. Tiene capacidad para comunicarse vía inalámbrica con cualquier nodo. Cuenta con conector de expansión para luz, temperatura, RH, Presión barométrica, acelerometro, acústico, magnético y otras tarjetas sensores MEMSIC. Las aplicaciones de MICAz son monitorización en interiores de construcciones y seguridad, acústica, vídeo, vibración.

TELOSB [37] funciona en el estándar IEEE 802.15.4. Tiene una velocidad alta de comunicación de radio a 250 kbps. Cuenta con el microcontrolador TI MSP430 con 10 kb de RAM. La antena se encuentra integrada a la tarjeta. Permite recolectar datos y programarse vía una interfaz USB. Utiliza un sistema operativo open source. Se encuentra integrado los sensores de temperatura, luz y humedad. Entre sus aplicaciones se encuentra la investigación del bajo consumo de energía y para experimentos en WSN.

IRIS [36] es un sistema para mediciones inalámbricas opera a 2.4 GHz en el estándar IEEE 802.15.4. Diseñado específicamente para redes de sensores embebidas. Su tasa de comunicación de datos es de 250 kbps. Cuenta con capacidad de enrutamiento permitiendo la comunicación inalámbrica con otros nodos. Cuenta con un conector de expansión para tarjetas de luz, temperatura, RH, presión ba-

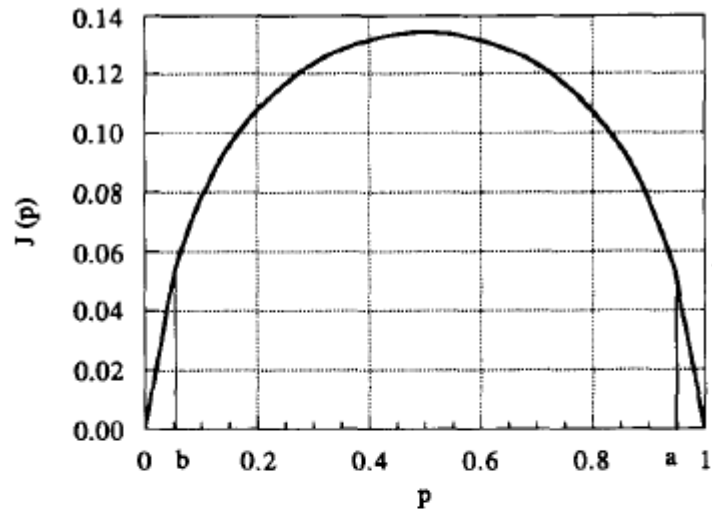


### 3.5. PLATAFORMAS

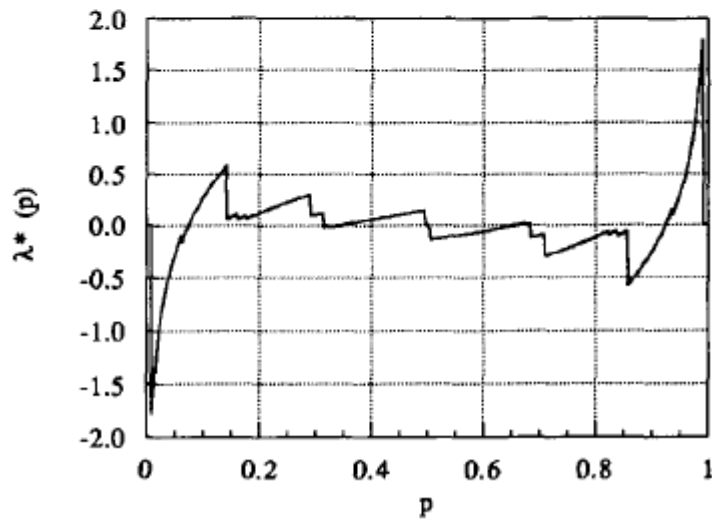
---

rométrica, aceleración, acústica, magnética y otras tarjetas sensoras MEMSIC. Entre sus aplicaciones se encuentra el monitoreo de interiores de construcciones y seguridad, acústica, vídeo, vibración.

LOTUS [34] plataforma que cuenta con un procesador Cortex M3 de 32 bit de 10 a 100 MHz. Cuenta con 64kB de SRAM, 512 kB FLASH, 64 MB de serial FLASH. Cuenta con antena integrada, utiliza el estándar IEEE 802.15.4. La velocidad alta de datos es de 250 kps. Cuenta con un LED indicador multicolor. Cuenta con una interfaz USB con conector tipo mini-B. Cuenta con una conector de expansión para tarjetas con sensores de luz, temperatura, RH, presión barométrica, acelerometro, acústica, magnético y otras tarjetas sensoras MEMSIC. Entre sus aplicaciones se encuentra el monitoreo industrial y de análisis, monitoreo de sismos y vibraciones, acústica, vídeo.

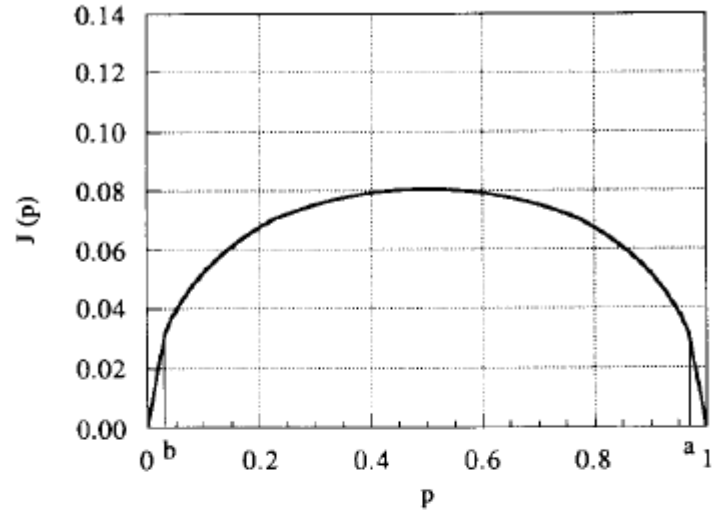


(a)

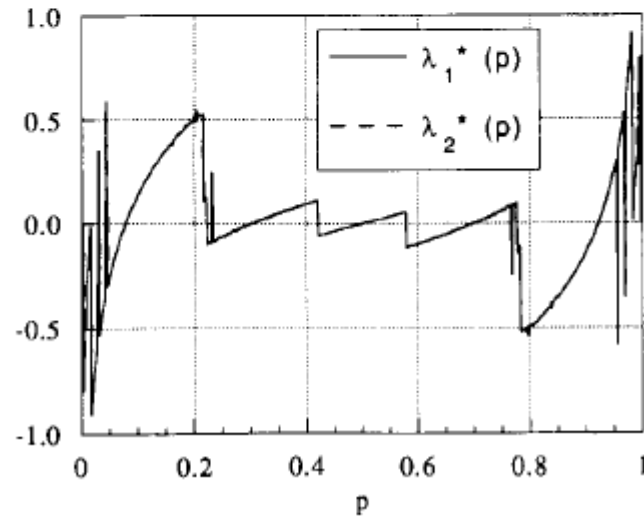


(b)

Figura 3.20: Resultados por el caso de un sensor con  $c = 0,01$ ,  $v = 1,0$ , y  $L_a = L_1 = 1,0$  a) Function Infinite-horizon b) Optimum stationary LDF thresholds [54]



(a)



(b)

Figura 3.21: Resultados por el caso de dos sensores idénticos con  $c = 0,01$ ,  $v = 1,0$ , y  $L_a = L_1 = 1,0$  a) Function Infinite-horizon b) Optimum stationary LDF thresholds [54]

### 3.5. PLATAFORMAS

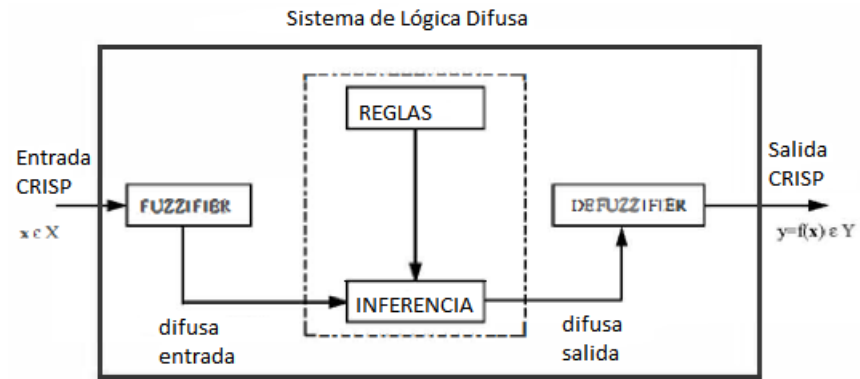


Figura 3.22: Estructura del sistema [46]

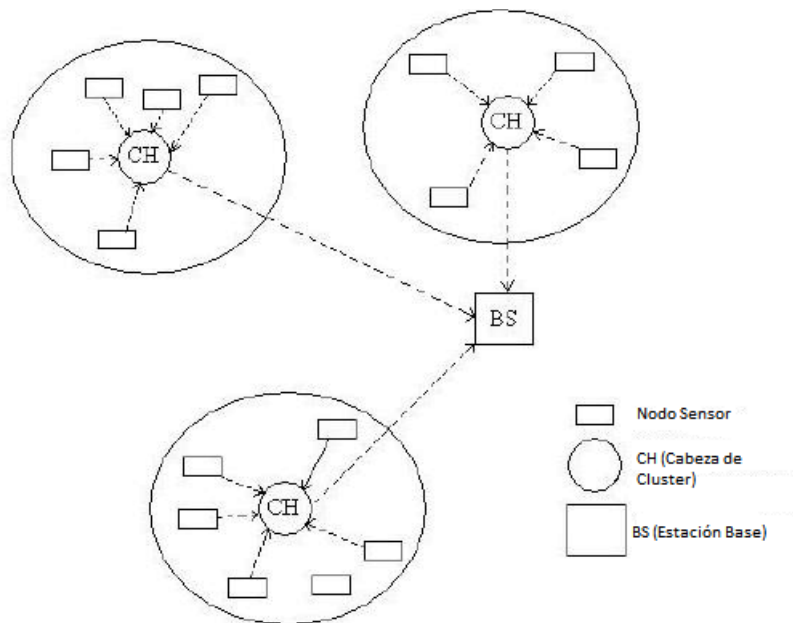


Figura 3.23: Ejemplo de configuración de una WSN [48]

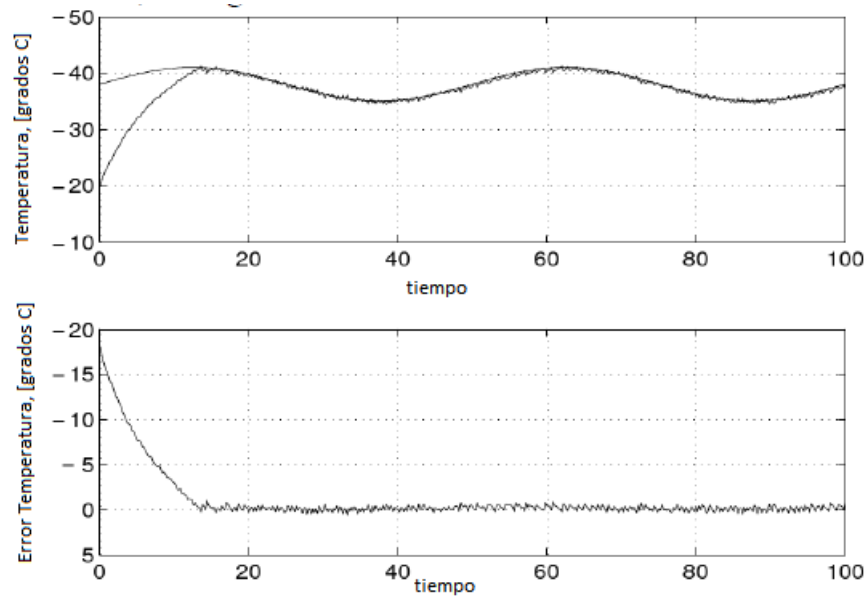


Figura 3.24: Respuesta de la temperatura [19]

## Capítulo 4

# TinyOS y NesC

### 4.1. TinyOS

TinyOS [20] es un sistema operativo diseñado específicamente para sensores inalámbricos con bajo consumo de energía. Se distribuye como código abierto<sup>1</sup> con licencia Berkeley Software Distribution (BSD) [3]. A diferencia de otros sistemas operativos, el diseño de TinyOS se enfoca en operaciones de ultra bajo consumo de energía. Su arquitectura está basada en componentes, permite una rápida implementación de aplicaciones al mismo tiempo que minimiza el tamaño del código requerido, algo que es de suma importancia debido a las severas limitaciones de memoria que existen en los nodos. La biblioteca de componentes de TinyOS incluye protocolos de red, servicios de distribución, manejadores de sensores y herramientas para la adquisición de datos, entre otros. TinyOS fue originalmente desarrollado como un proyecto de investigación de la Universidad de California en Berkeley, y a partir de entonces se ha conformado una gran comunidad internacional de desarrolladores y usuarios.

Para facilitar el desarrollo de aplicaciones, TinyOS proporciona un importante conjunto de servicios, componentes y abstracciones, tales como de comunicación y almacenamiento, así como para medir condiciones del ambiente. Además, se ejecuta en una gran cantidad de plataformas genéricas, siendo sencillo adaptarlo para que soporte nuevas plataformas.

TinyOS proporciona tres elementos de alto nivel con la finalidad de simplificar el desarrollo de sistemas y aplicaciones [29]:

- un modelo de componentes, que define cómo escribir pequeñas y reutilizables piezas de código, así como la manera de integrarlas para construir abstracciones de mayor tamaño;
- un modelo de ejecución concurrente, que define la manera en que los componentes pueden entrelazar sus ejecuciones, y la manera en que el código interactúa con interrupciones y sin interrupciones;

---

<sup>1</sup>(*Open Source*)

- interfaces de programación de aplicaciones (APIs), servicios, componentes de bibliotecas, así como una estructura general de componentes que simplifica la escritura de nuevas aplicaciones y servicios.

Las aplicaciones y sistemas en TinyOS, así como el sistema operativo, están escritos en lenguaje nesC [45].

## 4.2. nesC

nesC [18] es una extensión del lenguaje de programación C y ha sido diseñado para ajustarse a los conceptos estructurales y modelo de ejecución de TinyOS. Tiene como objetivos el desarrollo de aplicaciones que reduzcan el tamaño del código y de la utilización de memoria RAM, permitir la optimización en el uso de los recursos disponibles, así como ayudar en la prevención de errores de bajo nivel, tales como las condiciones de carrera.

Los conceptos básicos de nesC se explican a continuación [8]:

- Separación de construcción y composición: los programas son construidos a partir de componentes, los cuales son enlazados (“*wired*”) para formar programas. Los componentes son definidos en dos ámbitos, uno para la especificación (contiene los nombres de las instancias de las interfaces) y una para la implementación. Los componentes tienen concurrencia interna en forma de tareas (*tasks*). Los hilos de control pueden pasar en un componente a través de las interfaces. Estos hilos tienen sus raíces en una tarea o en una interrupción por hardware.
- Especificación del comportamiento de los componentes en términos del conjunto de interfaces. Las interfaces pueden ser proporcionadas o utilizadas por los componentes. Las interfaces buscan representar la funcionalidad que proporciona el componente al usuario. Las interfaces utilizadas representan la funcionalidad que el componente requiere para desempeñar su trabajo.
- Las interfaces son bidireccionales: especifican un conjunto de funciones que serán implementadas por el proveedor de la interfaz (comandos) y un conjunto de funciones que serán implementadas por el usuario de la interfaz (eventos). Esto permite que una interfaz represente una compleja interacción entre componentes (por ejemplo, el registro del interés en algún evento, seguido de una notificación cuando el evento ocurre). Esto es crítico debido a que todos los comandos de larga duración en TinyOS, como por ejemplo el envío de un paquete, son operaciones *sin bloqueo*; su terminación es notificada a través de un evento (envío realizado). Al especificar interfaces, un componente no puede invocar el comando `send` al menos que proporcione una implementación del evento `sendDone`.
- Los componentes están ligados estáticamente entre sí a través de sus interfaces. Esto incrementa la eficiencia en tiempo de ejecución, promueve un diseño robusto y permite un mejor análisis estático de los programas.

### 4.3. INSTALACIÓN Y CONFIGURACIÓN

---

- nesC ha sido diseñado bajo la premisa de que el código será generado por compiladores de programas completos. Esto también debe permitir una mejor generación de código y análisis del mismo.

### 4.3. Instalación y Configuración

En esta sección se explica el proceso de instalación de TinyOS en Ubuntu Lucid [1]. Para aprovechar el sistema de instalación de paquetes de Ubuntu, es recomendable agregar algún repositorio que contenga TinyOS. Por ejemplo, en el archivo de repositorios `/etc/apt/sources.list` se agrega la siguiente línea:

```
deb http://tinyos.stanford.edu/tinyos/dists/ubuntu lucid
main
```

Una vez hecho esto es necesario actualizar el listado de paquetes disponibles en los repositorios. Esto puede hacerse ejecutando desde una terminal lo siguiente:

```
$ sudo apt-get update
```

Para conocer cuál es la más reciente versión de TinyOS existente en el repositorio, puede escribirse:

```
$ apt-cache search tinyos
```

Suponiendo que el paquete con la versión más reciente sea `tinyos-2.1.1`, para instalarlo del repositorio se ejecuta en una terminal el siguiente comando:

```
sudo apt-get install tinyos-2.1.1
```

Una vez instalado TinyOS puede ser necesario hacer algunos ajustes en la configuración del sistema. Por ejemplo, es recomendable definir en las variables de entorno del sistema, el directorio en el que se encuentran los archivos importantes de TinyOS, tales como componentes, interfaces, bibliotecas, entre otros. Una manera de hacerlo consiste en agregar la siguiente línea en los archivos `/.bashrc` o `/.profile` que se encuentran en el directorio personal (*home directory*):

```
source /opt/tinyos-2.1.1/tinyos.sh
```

Para verificar si el entorno está configurado correctamente, debe ejecutarse:

```
$ tos-check-env
```

En algunos casos, el comando anterior puede generar un aviso de advertencia (*warning*) al no encontrar el archivo `tinyos.jar` en la ruta de clases (*classpath*). Para corregirlo, es necesario cambiar el valor de la variable `CLASSPATH` que se encuentra en el archivo `/opt/tinyos-2.x/tinyos.sh`. En nuestro caso, la ruta del archivo es `/opt/tinyos-2.1.1/support/sdk/java/tinyos.jar`, por lo que la variable de entorno fue modificada de la siguiente manera:



### 4.3. INSTALACIÓN Y CONFIGURACIÓN

---

```
export CLASSPATH=/opt/tinyos-2.1.1/support/sdk/java/
       /tinyos.jar:.
```

En caso de que se utilice Ubuntu para 64 bits es probable que la versión de Python incluida en la distribución sea diferente a la esperada. Por tal motivo, es necesario modificar el archivo `/opt/tinyos-2.1.1/support/make/sim.extra` en la línea `PYTHON_VERSION` para indicar la versión correcta. En nuestro caso, esta línea se modificó de la siguiente manera:

```
PYTHON_VERSION=2.6
```

Estas modificaciones son necesarias para poder ejecutar sin problemas el simulador de TinyOS TOSSIM.

Por otra parte, puede ocurrir que la versión utilizada de TinyOS no incluya el soporte para la base programadora utilizada, o que la información de ésta sea incorrecta en el sistema. Esto puede corregirse en ocasiones de la siguiente manera. Suponiendo que la base programadora a utilizar sea *mib520*, será necesario editar el archivo `/usr/bin/motelist` y corregir la sección en la que describen los parámetros de la base programadora. Por ejemplo, en nuestro caso, en el que se utilizó la base programadora *mib520*, fue necesario modificar la siguiente sección de texto:

```
#Scan /sys/bus/usb/drivers/usb for FTDI devices
    my @ftdidevs = grep { ($_->{UsbVendor}||"") eq
        "0403" && ($_->{UsbProduct}||"") eq "6001" }
```

por:

```
#Scan /sys/bus/usb/drivers/usb for FTDI devices      my @ftdidevs = grep
{ ($_->{UsbVendor}||"")eq "0403" && ($_->{UsbProduct}||"")eq "6010"
}
```

Para verificar que nuestro dispositivo es reconocido por TinyOS, es necesario escribir sobre una terminal:

```
$ motelist
```

Si después de estos cambios la base programadora es soportada por la versión del sistema operativo, el comando anterior mostrará el nombre y modelo de nuestro dispositivo.

## Capítulo 5

# Métodos propuestos para la detección de incendios forestales

### 5.1. Algoritmo para la detección de fuego basado en umbrales

Para poder definir el algoritmo para la detección de incendios, era importante en primer lugar conocer el comportamiento de las condiciones ambientales. Para esto, se midieron los valores de la temperatura, humedad relativa e intensidad de la luz por varios días durante los meses de Julio y Agosto, ya que son los meses con mayor incidencia de incendios en la región norte del estado de Baja California, México, y del sur del estado de California, Estados Unidos. Estas mediciones se hicieron sin la existencia de incendios. Además, se generaron artificialmente algunos incendios y se registraron los valores medidos por los sensores, con la finalidad de comparar estos valores con los obtenidos en condiciones normales. Los incendios fueron simulados utilizando una antorcha, como se muestra en la Fig. 5.1. Las características de la antorcha se eligieron buscando crear condiciones similares a las que existen al iniciar un incendio, ya que el algoritmo tiene como finalidad detectar los incendios en sus etapas iniciales. El algoritmo propuesto se definió tomando en consideración el análisis de la información recopilada.

Para la elaboración de la aplicación que registró las mediciones en los sensores y los envíos a la estación base se utilizó el lenguaje de programación nesC [45] (versión 1.2.4) y el sistema operativo TinyOS [3] (versión 2.1). Cada nodo está compuesto por una plataforma de hardware IRIS con una placa sensora modelo MTS420/400CC. La estación base está compuesta por una plataforma de hardware IRIS con un procesador ATmega1281, en una base programadora para la adquisición de datos modelo mib520.

## 5.1. ALGORITMO PARA LA DETECCIÓN DE FUEGO BASADO EN UMBRALES

---



Figura 5.1: Generación artificial de incendio

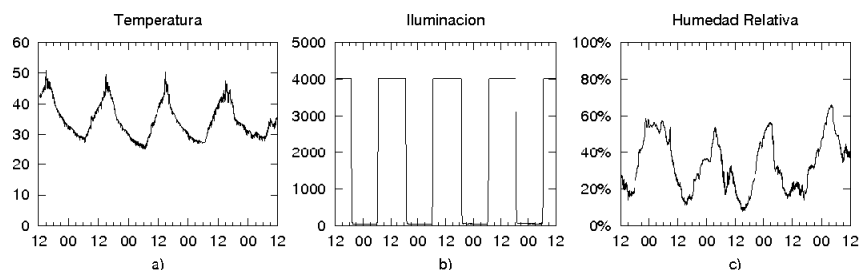


Figura 5.2: Mediciones en condiciones normales a) Temperatura, b) Luz visible, y c) Humedad relativa

Los nodos estuvieron midiendo los valores de temperatura, humedad relativa e intensidad de la luz durante varios días, en condiciones normales y en presencia de incendios artificialmente generados. Estos sensores fueron elegidos ya que están presentes en muchas plataformas WSN, y una de las motivaciones del proyecto ha sido mostrar que es posible detectar eficientemente la mayoría de los incendios forestales utilizando sensores económicos y de alta disponibilidad. Sin embargo, la utilización de sensores adicionales, como detectores de humo, cámaras de vídeo y de velocidad del viento mejorarían la eficiencia del sistema, a expensas de mayor costo y consumo de energía. Los sensores fueron instalados en árboles de una de las áreas de jardines del campus universitario, que presenta condiciones similares a los bosques de la región, y registraron los datos medidos durante los meses de Julio y Agosto de 2011. La información recopilada fue analizada empíricamente con la finalidad de identificar las características que permitan diferenciar las condiciones normales de las condiciones que prevalecen en un incendio. Las conclusiones obtenidas del análisis de los datos son las siguientes:

1. Las mediciones de las condiciones ambientales (temperatura, humedad relativa e intensidad de la luz) muestran un comportamiento cíclico durante el día, como puede observarse en la Fig. 5.2. Esta observación es impor-

## 5.1. ALGORITMO PARA LA DETECCIÓN DE FUEGO BASADO EN UMBRALES

---

tante ya que permite utilizar la correlación temporal entre las mediciones actuales y las mediciones de los días u horas previas.

2. Durante la noche, las mediciones de la intensidad de la luz se mantienen en un nivel constante muy bajo, mientras que durante el día se mantiene en un nivel constante muy alto. La transición de noche a día y viceversa genera un cambio drástico del nivel de intensidad. Este cambio súbito se da en un lapso menor a los 30 minutos.
3. La temperatura y la humedad mantienen una relación de sus valores inversamente proporcional. Esto es, cuando hay un incremento en la temperatura existe un decremento en la humedad, tanto en condiciones normales como en presencia de incendios.
4. El evento que provoca un incremento rápido en la temperatura y por lo tanto un decremento rápido en la humedad, es la exposición del mote a los rayos directos del sol.
5. Cuando ocurre un incendio, el valor de la temperatura incrementa drásticamente, mientras que el de la humedad decrementa de manera muy rápida. Cuando se utilizaron antorchas mas grandes si se detectó un cambio en los niveles de intensidad de la luz. Sin embargo, como se busca que la aplicación detecte el incendio en su etapa inicial, se utilizaron antorchas de menor tamaño.
6. Cuando el incendio ocurre en la noche las variaciones de la temperatura y humedad son similares a los incendios de día. Sin embargo, el incremento en la intensidad de la luz es significativo.
7. Para identificar un incendio de noche, puede utilizarse además de la temperatura, el valor de la intensidad de la luz. Si el cambio en éste último es muy grande, es muy probable que se trate de un incendio de noche. Es importante señalar que al amanecer también se tiene un rápido incremento en la intensidad de la luz. Sin embargo, en este caso la temperatura no muestra un incremento tan grande.
8. El incendio y la exposición directa a los rayos del sol provocan un drástico incremento en la temperatura. Sin embargo, el decremento en el valor de la humedad es mucho mayor en el caso de un incendio. Por lo tanto, para identificar un incendio de día y diferenciarlo de la exposición directa al sol, después de detectar un incremento rápido en la temperatura se revisará la razón de cambio en la humedad, así como el decremento registrado de la humedad relativa.

Con base en estas observaciones se definió un algoritmo para la detección de incendios en bosques.

El algoritmo propuesto utiliza un método de fusión de información denominado método del umbral [14]. El algoritmo está basado en la máquina de estados que se muestra en la Fig. 5.3, que considera seis estados. Los cambios

## 5.1. ALGORITMO PARA LA DETECCIÓN DE FUEGO BASADO EN UMBRALES

de un estado a otro se generan de acuerdo a las variaciones en los valores de temperatura, humedad relativa e intensidad de la luz.

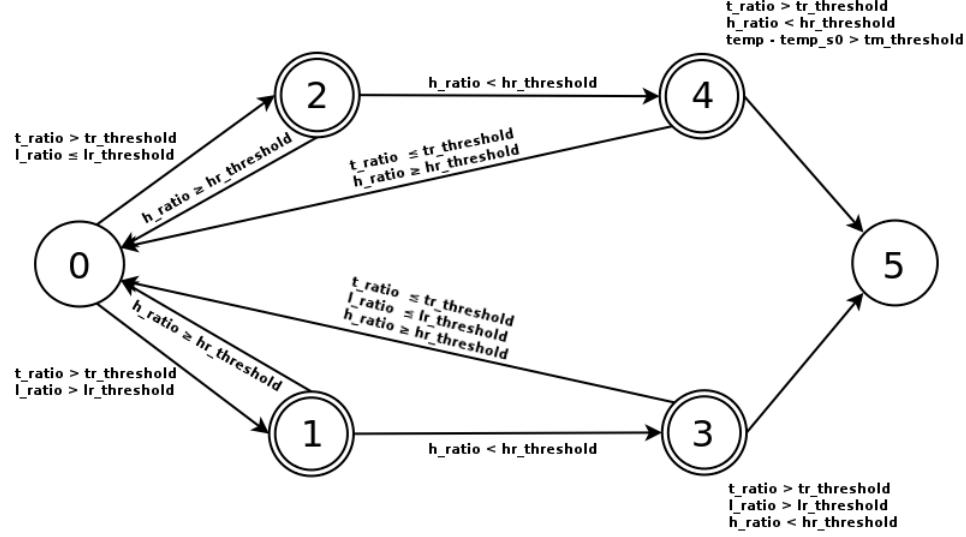


Figura 5.3: Máquina de estados utilizada por el algoritmo

Cada estado de la máquina de estados de la Fig. 5.3 representa una condición específica del entorno, y que tiene que ver con la probable existencia de un incendio. El Estado 0 es el estado inicial y representa al entorno en condiciones normales. Los estados Estado 1 y Estado 2 son de transición, ya que representan la ocurrencia del inicio de un evento, que puede ser un incendio de noche o un incendio de día, respectivamente. El Estado 3 puede representar la salida del sol (amanecer), mientras que el Estado 4 indica que el sensor fue expuesto directamente a los rayos del sol. El Estado 5 indica la presencia de un incendio.

En condiciones ambientales normales, el estado de la máquina es Estado 0. Esto significa que no hay algún incendio ni que el sensor está expuesto de manera directa a los rayos del sol. A pesar de que se registran los tres valores (temperatura, humedad relativa e iluminación), el único valor que se evalúa en este estado es el de la temperatura. En cada nueva medición, se calcula la razón (ratio) del promedio de los valores de la ventana deslizante de tamaño  $W_T$ , y el mas reciente valor de la temperatura. La ventana deslizante contiene los últimos  $W_T$  valores registrados de la temperatura. Si la razón es mayor que el umbral  $tr\_threshold$ , significa que la razón de cambio de la temperatura ha sido muy grande y por lo tanto algún evento ha ocurrido. Para identificar si el evento ha ocurrido de noche o de día, se revisa entonces la razón de cambio de la intensidad de la luz de manera similar a la de la temperatura. Por lo tanto, si la razón del promedio de los valores de la ventana deslizante de tamaño  $W_L$  es mayor que el umbral  $tl\_threshold$ , la maquina cambia al Estado 1. De otra manera, cambia al Estado 2. Es importante señalar que si la máquina deja el Estado 0, se almacena

## 5.2. TEORÍA DE FUSIÓN DE DATOS DEMPSTER-SHAFER

---

en la variable  $temp\_s0$  el ultimo valor estable de la temperatura. Esto es, se registra el valor de la temperatura previo al que genero el cambio de estado.

Si la máquina se encuentra en el Estado 1, se calcula la razón del promedio de los valores de la ventana deslizante de tamaño  $W_H$ , y el mas reciente valor de la humedad relativa. Si la razón es menor que el umbral  $th\_threshold$ , la maquina cambia al Estado 3; de otra manera, la maquina cambia al estado inicial Estado 0. En el Estado 3 evalúa de nuevo las razones de los tres valores y se comparan con sus respectivos umbrales. En caso de que las relaciones de evento persistan, la maquina cambia al Estado 5 y se emite la alarma de que un probable incendio de noche ha ocurrido.

Si la máquina cambió del Estado 0 al Estado 2, existe la posibilidad de:

1. que se haya generado un incendio
2. que el sensor haya sido expuesto directamente a los rayos del sol
3. que la temperatura se haya incrementado rápidamente en condiciones normales.

Para identificar lo ocurrido, se calcula la razón del promedio de los valores de la ventana deslizante de tamaño  $W_L$ , y el mas reciente valor de la humedad relativa. Si la razón es menor que el umbral  $th\_threshold$ , la máquina cambia al Estado 4; de otra manera, la máquina cambia al estado inicial Estado 0. El Estado 4 puede representar la exposición del sensor a los rayos directos del sol. Si en este estado, la razón de la temperatura actual y el promedio de la ventana deslizante de tamaño  $W_T$  sigue siendo mayor que el umbral  $tr\_threshold$ , y la razón del promedio de los valores de la ventana deslizante de tamaño  $W_H$  y el mas reciente valor de la humedad relativa sigue siendo menor que el umbral  $th\_threshold$ , se calcula la diferencia del valor de la temperatura actual  $temp$  y el valor de  $temp\_s0$ . Si la diferencia es mayor que el umbral  $tm\_threshold$ , significa que el incremento en la magnitud de la temperatura ha sido muy grande, por lo que la máquina cambia al Estado 5 y emite una alarma de que un probable incendio se ha generado.

Para la detección de fuego en bosques se diseñaron y evaluaron dos métodos el primero basado en umbrales y el segundo utilizando la metodología Dempster-Shafer. Para el primer método se definió una máquina de estados en la cual se indica el tipo de evento que esta ocurriendo para indicar si existe fuego o representa condiciones normales, en el segundo método se definió un diagrama de flujo el cual realiza la fusión de masas probabilistas entre dos estados considerados ambiente con fuego y ambiente sin fuego, la normalización de la fusión de datos mediante el método Dempster-Shafer [27] indicara si existe fuego en el ambiente o no existe fuego en el ambiente.

## 5.2. Teoría de fusión de datos Dempster-Shafer

La fusión de datos se puede definir como la combinación de datos producidos por uno o mas sensores de una manera que da una mejor estimación con la

## 5.2. TEORÍA DE FUSIÓN DE DATOS DEMPSTER-SHAFER

---

cantidad que estamos midiendo [27].

El modelo básico de la fusión de datos define la relación entre las fuentes de datos y los tipos de procesamiento que podría llevarse a cabo para extraer la máxima información posible de el. Entre la fuente de datos y el humano, existen varios niveles de procesamiento, los cuales son [11]:

- Preprocesamiento de la fuente: aquí se crea la información preliminar de los datos que sirven para una mejor comunicación con otros niveles de procesamiento.
- Nivel 1 Refinamiento de Objetos:
  - Primero, se transforman los datos de los sensores con unidades consistentes.
  - Segundo, se refina y extiende las estimaciones
  - Tercero, se asignan los datos a los objetos para permitir a la aplicación una estimación estadística.
  - Finalmente, se refina la estimación de la identidad del objeto o clasificación.
- Nivel 2 Refinamiento de la Situación: se procesa el desarrollo de la descripción de las relaciones entre los objetos y eventos en los contextos del ambiente.
- Nivel 3 Refinamiento de la amenaza: se procesan los proyectos de la situación actual en el futuro para obtener conclusiones acerca de amenazas, vulnerabilidades y oportunidades de operaciones.
- Nivel 4 Proceso de refinamiento: en este nivel no se preocupa mucho sobre los datos, sino mas bien se preocupa por lo que están haciendo los demás niveles y si lo puede optimizar.
- Administración de los datos: la colección de funciones provee el acceso, administración, bases de datos, almacenamiento, compresión, consultas relacionales y protección de los datos. Es la tarea mas crucial, especialmente si se trata con grandes cantidades de datos o cálculos complejos.

La teoría Dempster-Shafer se enfoca en la matemática de la evidencia [49]. En la teoría Dempster-Shafer, la evidencia puede ser asociada con varios posibles eventos. Donde la evidencia es suficiente para asignar probabilidades a eventos únicos, el modelo Dempster-Shafer se convierte en una tradicional formulación probabilística.

Con el método Dempster-Shafer se asigna un valor de masa a cada conjunto perteneciente al sistema.

La regla de Normalización Dempster para la fusión de dos masas tiene la siguiente formula:

### 5.3. MÉTODO BASADO EN LA TEORÍA DEMPSTER-SHAFER PARA LA DETECCIÓN DE FUEGO

---

$$m^{1,2}(C) = \frac{\sum_{A \cap B = C} m^1(A)m^2(B)}{\sum_{A \cap B \neq \emptyset} m^1(A)m^2(B)} = \frac{\sum_{A \cap B = C} m^1(A)m^2(B)}{1 - \sum_{A \cap B = \emptyset} m^1(A)m^2(B)} \quad (5.1)$$

donde estas masas son fusionadas usando la regla de combinación Dempster:

$$m^{1,2}(C) \propto \sum_{A \cap B = C} m^1(A)m^2(B) \quad (5.2)$$

### 5.3. Método basado en la teoría Dempster-Shafer para la detección de fuego

Para el procesamiento de la información en la metodología Dempster-Shafer se debe contar con los valores de las masas probabilísticas de los estados que se desea clasificar, debido a que las masas probabilísticas deben ser obtenidas con anterioridad (nivel 1), se propone calcular las masas probabilísticas utilizando la interpolación de Lagrange y formulación de Newton con diferencias divididas para su posterior fusión, con lo que se estimará la identidad del evento, el refinamiento de la situación (nivel 2) sera llevado a cabo con la relación entre los diferentes valores de las masas probabilísticas en los estados definidos. Los estados considerados son con presencia de fuego y con ausencia de fuego.

### 5.4. Interpolación Lineal

De manera general, en la interpolación lineal se utilizan dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , para obtener un tercer punto interpolado  $(x, y)$  a partir de la formula siguiente:

$$y = y_1 + (x - x_1) \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (5.3)$$

donde la pendiente  $m$  entre los dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$  es:

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (5.4)$$

### 5.5. Interpolación de Lagrange

La definición de la Interpolación de Lagrange viene dada por la ecuación (5.5):

$$P_n(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + \dots + y_n L_{n,n}(x) = \sum_{i=0}^n y_i L_{n,i}(x) \quad (5.5)$$



## 5.6. FORMULACIÓN DE NEWTON CON DIFERENCIAS DIVIDIDAS

---

donde  $L_i(x)$  es un polinomio de grado  $n$ :

$$L_{n,i}(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}, i = 0, 1, \dots, n$$

(5.6)

se tiene:

$$\begin{aligned} P_n(x_0) &= y_0 \\ P_n(x_1) &= y_1 \\ &\vdots \\ P_n(x_n) &= y_n \end{aligned}$$

### 5.6. Formulación de Newton con diferencias divididas

El polinomio interpolador de diferencias divididas de Newton tiene la forma que indica la formula (5.7):

$$P_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \quad (5.7)$$

donde:

$$\begin{aligned} a_0 &= f[x_0] \\ a_1 &= f[x_0, x_1] \\ a_2 &= f[x_0, x_1, x_2] \\ &\vdots \\ a_n &= f[x_0, x_1, \dots, x_n] \end{aligned}$$

las diferencias divididas se definen como:

## 5.7. DIAGRAMA DE FLUJO PARA LA DETECCIÓN DE FUEGO

$$\begin{aligned}
 f[x_i] &= y_i \\
 f[x_i, x_{i+1}] &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \\
 f[x_i, x_{i+1}, x_{i+2}] &= \frac{\frac{(y_{i+2} - y_{i+1})}{(x_{i+2} - x_{i+1})} - \frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)}}{(x_{i+2} - x_i)} \\
 &\vdots \\
 f[x_i, x_{i+1}, \dots, x_{i+k}] &= \frac{f[x_i, x_{i+1}, \dots, x_{i+k-1}] - f[x_{i+1}, \dots, x_{i+k}]}{x_{i+k} - x_i}
 \end{aligned}$$

## 5.7. Diagrama de flujo para la detección de fuego

El método para la detección de fuego sigue el diagrama de flujo de la Fig. 5.4. Para iniciar se necesita el valor de la temperatura medida *TempMedida* y el valor de la humedad medida *HumMedida*, el procedimiento de asignación y cálculo de masas probabilísticas en estados se describe en la sección 5.8, la fusión de masas y normalización se describe en la sección 5.9, en base a lo que se obtiene el la normalización se toma una decisión, la cual consiste enviar un mensaje indicando la presencia o ausencia de fuego.

## 5.8. Cálculo de Masas

El valor probabilístico de masa individual sera representado como se muestra a continuación:

$$masa_{Tipo\ de\ Sensor}^{Metodo\ de\ Interpolacion}$$

En la parte del superíndice de la masa individual se escribe la letra del método de interpolación correspondiente y en la parte del subíndice se escribe la letra del tipo de sensor correspondiente. La representación de los superíndices se encuentran en el Cuadro 5.1 y la representación de los subíndices en el Cuadro 5.2.

Metodo de Interpolacion	Letra
Interpolacion de Lagrange	L
Formulacion de Newton con diferencias divididas	N
Metodo de interpolacion	I

Cuadro 5.1: Representación de los superíndices

La masa en estados correspondiente considerando los estados indicados en el cuadro 5.3 se representa como se muestra a continuación:

## 5.8. CÁLCULO DE MASAS

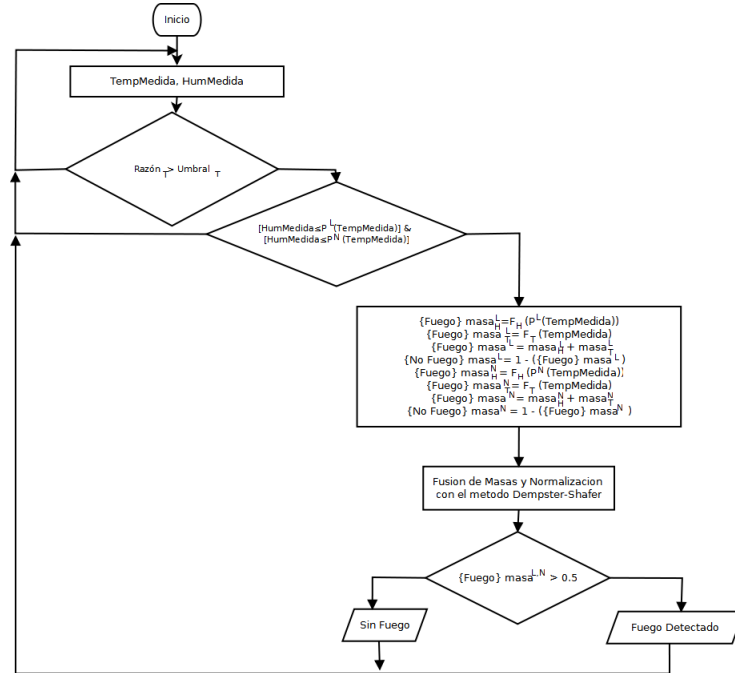


Figura 5.4: Diagrama De Flujo para la Detección de Fuego

$\{estado\}masa$

Las funciones polinómicas obtenidas serán representadas como se muestra a continuación:

$P_{Metodo\ de\ Interpolacion}$

En la parte del superíndice de  $P$  se escribe la letra correspondiente al método de interpolación como indica el Cuadro 5.1.

Las funciones de interpolación lineal obtenidas serán representadas como se muestra a continuación:

$F_{Tipo\ de\ sensor}$

Tipo de Sensor	Letra
Temperatura	T
Humedad	H
Cualquiera de los anteriores	S

Cuadro 5.2: Representación de los subíndices

## 5.8. CÁLCULO DE MASAS

---

Descripción	Estado
Cuando existe la presencia de fuego	Fuego
Estado que indica la ausencia de fuego	NoFuego
Cualquier estado	E

Cuadro 5.3: Estados

En la parte del subíndice de  $F$  se escribe la letra correspondiente al tipo de sensor utilizado como indica el cuadro 5.2 en la página anterior.

Para llevar a cabo la asignación de medidas para el posterior cálculo de las masas probabilísticas se sigue el Pseudocódigo 1. Donde los valores de entrada son la temperatura medida ( $TempMedida$ ) y la humedad medida ( $HumMedida$ ). En la línea 1 se verifica que la razón del promedio de la ventana deslizante en base a la temperatura ( $Razon_T$ ) se mayor al umbral del promedio de la razón del promedio de la ventana deslizante en base a la temperatura ( $UmbralProm_T$ ). En la línea 2 se compara la humedad medida con la humedad esperada en la función polinómica obtenida de Lagrange y la humedad medida con la humedad esperada en la función polinómica obtenida de Newton, la función polinómica se obtiene con los valores obtenidos en experimentos con presencia de fuego realizados con anterioridad utilizando la ecuación (5.5) y la ecuación (5.7), respectivamente, donde el valor de  $x$  corresponde a los valores obtenidos de la temperatura y los valores de  $y$  le corresponden los valores obtenidos de la humedad.

---

**Pseudocódigo 1** Asignación de medidas para posterior cálculo de masas probabilísticas

---

**Entrada:** Los valores  $TempMedida$ ,  $HumMedida$ .

- 1: **si**  $Razon_T > Umbral_T$  **entonces**
  - 2:   **si** ( $HumMedida \leq P^L(TempMedida)$ ) and ( $HumMedida \leq P^N(TempMedida)$ ) **entonces**
  - 3:     Calcular los valores de masa probabilística correspondiente a Interpolación de Lagrange y Newton
  - 4:   **si no**
  - 5:     Esperar nuevas mediciones
  - 6:   **fin si**
  - 7: **fin si**
- 

Para llevar a cabo el cálculo de los valores de masa probabilística correspondiente a la interpolación de Lagrange y Newton como indica la línea 3 se sigue con el pseudocódigo 2 para  $I = \{L, N\}$  con lo que se observa que se utiliza la ecuación (5.8) para obtener la masa probabilística en estado fuego para Lagrange y la ecuación (5.12) para obtener la masa probabilística en estado con fuego para Newton, donde  $masa_H^L$  se obtiene de la ecuación (5.9),  $masa_T^L$  se obtiene de la ecuación (5.10),  $masa_H^N$  se obtiene de la ecuación (5.13) y  $masa_T^N$  se obtiene de la ecuación (5.14). Las masas para los estados NoFuego corres-

## 5.8. CÁLCULO DE MASAS

---

pendientes en Lagrange y Newton se obtienen de las ecuaciones (5.11) y (5.15) respectivamente.

$$\{Fuego\}masa^L = masa_H^L + masa_T^L \quad (5.8)$$

$$masa_H^L = F_H(P^L(TempMedida)) \quad (5.9)$$

$$masa_T^L = F_T(TempMedida) \quad (5.10)$$

$$\{NoFuego\}masa^L = 1 - (\{Fuego\}masa^L) \quad (5.11)$$

$$\{Fuego\}masa^N = masa_H^N + masa_T^N \quad (5.12)$$

$$masa_H^N = F_H(P^N(TempMedida)) \quad (5.13)$$

$$masa_T^N = F_T(TempMedida) \quad (5.14)$$

$$\{NoFuego\}masa^N = 1 - (\{Fuego\}masa^N) \quad (5.15)$$

La función  $F$  de cada tipo de sensor se obtiene de la interpolación correspondiente, utilizando la ecuación (5.3) con los datos que muestra el Cuadro 5.4. Donde  $\min(masa)$  es el valor de la masa individual probabilística mínima asignada a cada tipo de sensor y  $\max(masa)$  tendrá el valor máximo de masa individual probabilística para cada tipo de sensor. La suma de las masas individuales será menor o igual a 100 %. Si con la interpolación utilizando  $F$  con su respectivo tipo de sensor se obtiene una masa individual mayor a  $\max(masa)$  correspondiente, se asigna el valor de masa individual  $\max(masa)$  correspondiente a su tipo de sensor, haciendo  $F_S = \max(masa_S)$ .

---

### Pseudocódigo 2 Cálculo de los valores de $masa^I$

---

**Entrada:** Los valores TempMedida y LuzMedida.

- 1:  $masa_H^I = F_H(P^I(TempMedida))$
  - 2:  $masa_T^I = F_T(TempMedida)$
  - 3: **si**  $masa_H^I > \max(masa_H)$  **entonces**
  - 4:    $masa_H^I \leftarrow \max(masa_H)$
  - 5: **fin si**
  - 6: **si**  $masa_T^I > \max(masa_T)$  **entonces**
  - 7:    $masa_T^I \leftarrow \max(masa_T)$
  - 8: **fin si**
  - 9:  $\{Fuego\}masa^I \leftarrow masa_H^I + masa_T^I$
  - 10:  $\{NoFuego\}masa^I \leftarrow 1 - (\{Fuego\}masa^I)$
-

## 5.9. FUSIÓN DE MASAS Y NORMALIZACIÓN

Masa Individual	$x_1$	$x_2$	$y_1$	$y_2$
$F_H$	<i>HumMediaInicial</i>	<i>HumMediaFinal</i>	$\min(masa_H)$	$\max(masa_H)$
$F_T$	<i>TempMediaInicial</i>	<i>TempMediaFinal</i>	$\min(masa_T)$	$\max(masa_T)$
TOTAL				100 %

Cuadro 5.4: Masa probabilística individual

## 5.9. Fusión de Masas y Normalización

Para determinar si existe presencia de fuego se fusionan las masas obtenidas anteriormente considerando los siguientes dos estados:

1. El estado “fuego” donde se indica que esta sucediendo un incendio.
2. El estado “No fuego” donde se indica la ausencia de fuego.

Se considera las dos masas obtenidas, en la quinta columna del cuadro 5.5 se muestra la fusión de las masas normalizado final calculada utilizando la ecuación (5.1). Las masas son fusionadas utilizando la regla de combinación Dempster. Esta regla en primera instancia es escrita como una simple proporcionalidad, usando la notación definida en la ecuación (5.2), correspondiente a la cuarta columna del cuadro 5.5.

Tipo	$masa^L$	$masa^N$	Combinacion $masa^{L,N}$	Normalizacion $masa^{L,N}$
Fuego				
No Fuego				
Masa Total	100 %	100 %		100 %

Cuadro 5.5: Asignación de Masas

Cuando se realiza el calculo normalizado se toma una decisión indicando que existe fuego cuando el valor del tipo fuego normalizado es mayor al 80 %.

## 5.10. Ejemplo del Método de Detección

Para la detección del fuego se deben tener de manera previa el valor de la *TempMediaInical*, *TempMediaFinal*, *HumMediaInicial*, *HumMediaFinal*, las funciones polinómicas de Lagrange y Newton que describen el fuego, donde la variable dependiente es la humedad y la variable independiente es la temperatura. Si se considera:

$$x = temperatura$$

#### 5.10. EJEMPLO DEL MÉTODO DE DETECCIÓN

---

$$y = \text{humedad}$$

$$\text{TempMediaInicial} = 37,8$$

$$\text{TempMediaFinal} = 45,8$$

$$\text{HumMediaInicial} = 28,8$$

$$\text{HumMediaFinal} = 23$$

$$\text{Razon}_T = 1,0197$$

$$\text{Umbral}_T = 1,01$$

$$\min(\text{masa}_H) = 30 \%$$

$$\max(\text{masa}_H) = 50 \%$$

$$\min(\text{masa}_T) = 30 \%$$

$$\max(\text{masa}_T) = 50 \%$$

$$P^L(x) = -\frac{35}{2958}x^2 + \frac{1415}{5916}x + \frac{363493}{9860}$$

$$P^N(x) = -\frac{23}{1188}x^2 + \frac{1121}{1485}x + \frac{61417}{2160}$$

Con los datos anteriores se puede obtener la función para los cálculos de las masas individuales  $\text{masa}_T$  y  $\text{masa}_H$ , utilizando la ecuación 5.3, considerando las variables  $x_0 = \text{TempMediaInicial}$ ,  $x_1 = \text{TempMediaFinal}$ ,  $y_0 = \min(\text{masa}_T)$ ,  $y_1 = \max(\text{masa}_T)$  para  $\text{masa}_T$  y las variables  $x_0 = \text{HumMediaInicial}$ ,  $x_1 = \text{HumMediaFinal}$ ,  $y_0 = \min(\text{masa}_H)$ ,  $y_1 = \max(\text{masa}_H)$  para  $\text{masa}_H$ , con lo que se obtiene las dos masas individuales, como sigue:

## 5.10. EJEMPLO DEL MÉTODO DE DETECCIÓN

---

$$x = temperatura$$

$$masa_T(x) = 30 + (x - 37,8) \frac{(50 - 30)}{(45,8 - 37,8)}$$

$$masa_T(x) = 30 + (x - 37,8) \frac{(5)}{(2)}$$

$$x = humedad$$

$$masa_H(x) = 30 + (x - 28,8) \frac{(50 - 30)}{(23 - 28,8)}$$

$$masa_H(x) = 30 - (x - 28,8) \frac{(100)}{(29)}$$

Con los datos anteriores se considera el caso siguiente, donde:

$$TempMedida = 38$$

$$HumMedida = 27,5$$

Siguiendo el diagrama de flujo de la Fig. 5.4, se observa que se cumple la primera condición  $Razon_T > Umbral_T$  ya que 1.0197 es mayor o igual a 1.01, después se verifica la condición  $(HumMedida \leq P^L(TempMedida))$  &  $(HumMedida \leq P^N(TempMedida))$ , para ello es necesario primero obtener  $P^L(TempMedida)$  y  $P^N(TempMedida)$ , como sigue:

$$P^L(38) = -\frac{35}{2958}(38)^2 + \frac{1415}{5916}(38) + \frac{363493}{9860}$$

$$P^L(38) \approx 28,8684584179$$

$$P^N(38) = -\frac{23}{1188}(38)^2 + \frac{1121}{1485}(38) + \frac{61417}{2160}$$

$$P^N(38) \approx 29,1630892256$$

Debido a que 27.5 es menor igual a 28.86 y 27.5 es menor igual a 29.16, se sigue con la asignación de masas individuales utilizando  $masa_T(x)$  donde  $x$  corresponde a la temperatura y en  $masa_H(x)$   $x$  corresponde a la humedad<sup>1</sup>. Los cálculos para las masas individuales, son:

---

<sup>1</sup>(no confundir las variables independientes  $x$  de las masas individuales, con las variables independientes  $x$  en las funciones polinómicas  $P^L(x)$  y  $P^N(x)$ )



### 5.10. EJEMPLO DEL MÉTODO DE DETECCIÓN

---

$$\begin{aligned}
\{Fuego\}masa_H^L(28,86) &= 30 - (28,86 - 28,8) \frac{100}{29} \\
\{Fuego\}masa_H^L(28,86) &\approx 29,7931034483 \\
\{Fuego\}masa_T^L(38) &= 30 + (38 - 37,8) \frac{5}{2} \\
\{Fuego\}masa_T^L(38) &= 30,5 \\
\{Fuego\}masa_H^N(29,16) &= 30 - (29,16 - 28,8) \frac{100}{29} \\
\{Fuego\}masa_H^N(29,16) &\approx 28,7586206897 \\
\{Fuego\}masa_T^N(38) &= 30,5
\end{aligned}$$

Después se calcula el valor de masa para el estado *Fuego*, utilizando la ecuación (5.8) para  $\{Fuego\}masa^L$  y utilizando la ecuación (5.12) para  $\{Fuego\}masa^N$ , como sigue:

$$\begin{aligned}
\{Fuego\}masa^L &= 29,79 + 30,5 = 60,29 \% \\
\{Fuego\}masa^N &= 28,75 + 30,5 = 59,25 \%
\end{aligned}$$

Ya que se tiene los valores de masa en el estado *Fuego*, se calculan los valores de masa en el estado *NoFuego*. Para  $\{NoFuego\}masa^L$  se utiliza la ecuación (5.11) y para  $\{NoFuego\}masa^N$  se utiliza la ecuación (5.15), calculando como sigue:

$$\begin{aligned}
\{NoFuego\}masa^L &= 100 - 60,29 = 39,71 \% \\
\{NoFuego\}masa^N &= 100 - 59,25 = 40,75 \%
\end{aligned}$$

Lo siguiente es realizar la fusión de masas y normalización, para ello se toman los valores calculados  $\{Fuego\}masa^L$ ,  $\{Fuego\}masa^N$ ,  $\{NoFuego\}masa^L$  y  $\{NoFuego\}masa^N$  y se colocan en la tabla mostrada en el cuadro 5.5, como sigue (ver Cuadro 5.6):

Para calcular los valores de la cuarta columna se utiliza la ecuación (5.2) y para la quinta se utiliza la ecuación (5.1). Debido a que 68.82 % es mayor al 50 % se concluye que existe presencia de fuego.

### 5.10. EJEMPLO DEL MÉTODO DE DETECCIÓN

---

Tipo	$masa^L$	$masa^N$	Combinación $masa^{L,N}$	Normalización $masa^{L,N}$
Fuego	60.29	59.25	35.72	68.82
No Fuego	39.71	40.75	16.18	31.17
Masa Total	100 %	100 %		$\sim 100 \%$

Cuadro 5.6: Cálculos de fusión y normalización

## Capítulo 6

# Evaluación

Para la llevar a cabo la evaluación de los métodos para la detección de fuego se programaron los motes para que realicen mediciones cada 7 segundos tomando los valores de la temperatura, humedad y luz visible, enviándolos en un paquete de datos cada 28 segundos a la estación base donde se almacenan en una base de datos en MySQL. El formato que tiene el paquete de datos que envía el mote a la estación base es como se muestra en la Fig. 6.1. En el paquete de datos en el cuerpo del mensaje se envía el id del mote (IDNODE), junto con cuatro mediciones de Temperatura (Temp\_data[4]), cuatro mediciones de la Humedad Relativa (Hum\_data[4]) y cuatro mediciones de la luz visible (VisLight\_data[4]).

AMPacket	Dirección Destino	Dirección Fuente	Tamaño Mensaje	ID del Grupo	ID del Manejador	NodeID	Temp_data[4]	Hum_data[4]	VisLight_data[4]
1 byte	2 bytes	2 bytes	1 byte	1 byte	1 byte	2 bytes	8 bytes	8 bytes	8 bytes

Figura 6.1: Formato Paquete de Datos

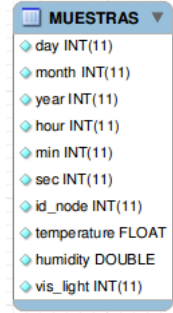
El paquete de datos que llega a la base datos es almacenado en una tabla llamada MUESTRAS que tiene la estructura como muestra la Fig. 6.2. Donde se registra el día, mes, año, hora, minuto, segundos cuando se hizo la medición en el campo day, month, year, hour, min, sec respectivamente; el id del mote es registrado en el campo id\_node; la temperatura se registra en grados centígrados en el campo temperature, la humedad relativa es registrada en el campo humidity, la luz visible es registrada en el campo vis\_light.

Las métricas para evaluar el desempeño del algoritmo y el método fueron dos:

1. Falsos positivos: cuando el algoritmo indica erróneamente que un incendio ha ocurrido.
2. Falsos negativos: cuando el algoritmo no detecta que un incendio ha ocurrido.

## 6.1. ALGORITMO PARA LA DETECCIÓN DE FUEGO BASADO EN UMBRALES

---



MUESTRAS	
day	INT(11)
month	INT(11)
year	INT(11)
hour	INT(11)
min	INT(11)
sec	INT(11)
id_node	INT(11)
temperature	FLOAT
humidity	DOUBLE
vis_light	INT(11)

Figura 6.2: Formato Tabla MUESTRAS

### 6.1. Algoritmo para la detección de fuego basado en umbrales

Como se mencionó previamente, se midieron los valores de la temperatura, humedad relativa e intensidad de la luz por varios días durante los meses de Julio y Agosto de 2011, tanto en condiciones normales como ante la presencia de incendios generados artificialmente. Las mediciones se hicieron cada siete segundos. Por consideraciones de ahorro de energía, los datos se enviaron a la estación base cada 28 segundos. En cada experimento, la antorcha se colocó a 10 cm del sensor durante un periodo de 10 minutos. Los experimentos se llevaron a cabo en diferentes días y horarios.

La definición de los parámetros utilizados en el algoritmo se hizo de manera empírica, y fueron resultado del análisis de los datos registrados. Los valores de las ventanas deslizantes  $W_T$ ,  $W_H$  y  $W_L$  variaron en el rango de 5 a 30, con incrementos de cinco en cinco, y se evaluó el desempeño de la aplicación con cada uno de estos valores.

Con los datos registrados, tanto de condiciones normales como con los incendios, se ejecutó el algoritmo propuesto utilizando diferentes valores de los tamaños de las ventanas deslizantes y de los umbrales. El tamaño de las ventanas deslizantes que generaron el mejor desempeño del algoritmo fue de 20, mientras que los mejores valores de los umbrales fueron:  $tr\_threshold = 1,01$ ,  $tl\_threshold = 1,1$ ,  $th\_threshold = 1$ , y  $tm\_threshold = 3^{\circ}C$ .

Ventana	Experimentos	Detectados	Falsos Positivos	Falsos Negativos
15	6	5	3	1
20	6	6	3	0
25	6	6	3	0
30	6	6	2	0
35	6	6	2	0

Cuadro 6.1: Resultados de algoritmo basado en umbrales

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

Como resultado de la evaluación del algoritmo (ver Cuadro 6.1), se obtuvo lo siguiente:

1. El algoritmo fue capaz de detectar la totalidad de los incendios generados. Por lo tanto, no se generaron falsos negativos.
2. El algoritmo generó algunos falsos positivos, que coincidieron con eventos en los que el sensor fue expuesto directamente a los rayos solares. Al analizar detalladamente estos casos, se encontró que los valores de la temperatura, humedad relativa e intensidad de la luz son muy similares a los que se registran en un incendio. Una alternativa para reducir la incidencia de falsos positivos consiste en cubrir a los nodos para evitar la exposición directa a la luz solar.

Es importante señalar que los valores de los umbrales deberán ajustarse tomando en consideración las características del entorno en el cuál se utilizará el algoritmo, y de la época del año. Para este fin sería de mucha utilidad contar con registros históricos de los valores de temperatura y humedad relativa del área en los que se desplegarán los sensores.

## 6.2. Método para detección de Fuego basado en Dempster-Shafer

Para la realización de la evaluación se considero los resultados obtenidos con el algoritmo basado en umbrales. Por lo que se utilizan únicamente los sensores de la temperatura y la humedad relativa, además como los problemas por la noche son de detección con el sensor de luz, se considera evaluar únicamente experimentos con presencia de luz. El número de experimentos realizados el día 28 de Julio de 2011 con presencia de luz son nueve. Las mediciones se realizaron desde las 9:42 hasta 19:20 horas del mismo día. Las gráficas de la relación entre la temperatura y humedad de los experimentos son las siguientes:

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

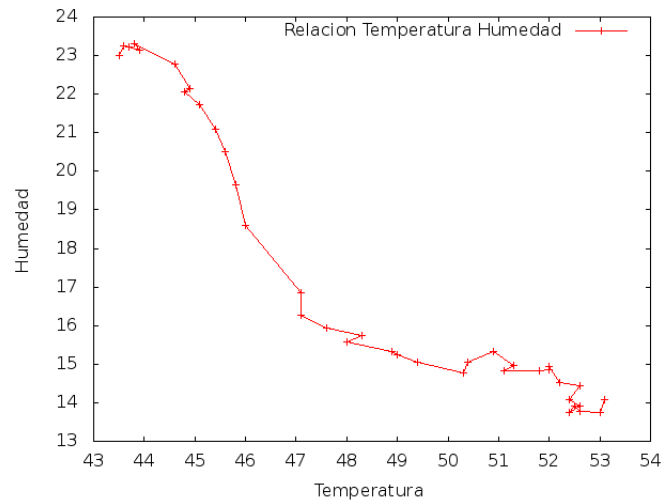


Figura 6.3: Experimento de 9:47 a 9:52 horas

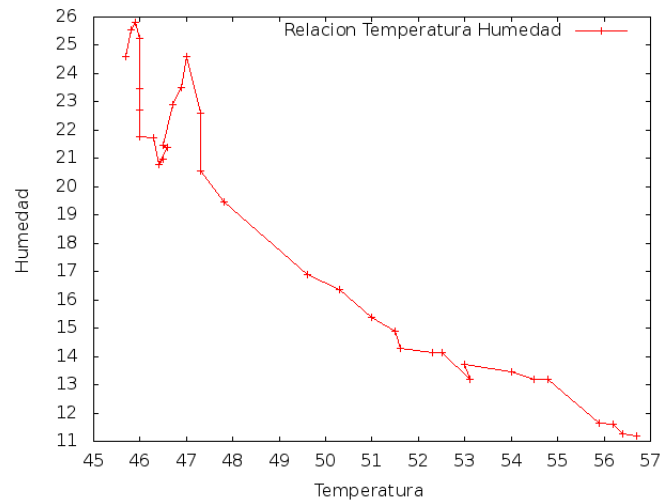


Figura 6.4: Experimento de 9:59 a 10:04 horas

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

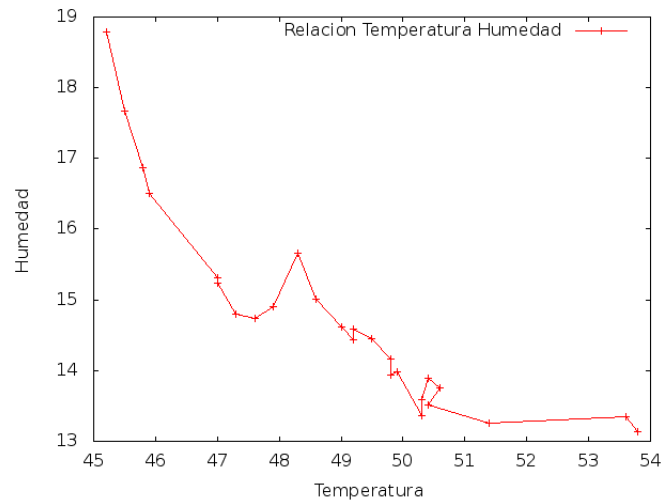


Figura 6.5: Experimento de 12:22 a 12:27 horas

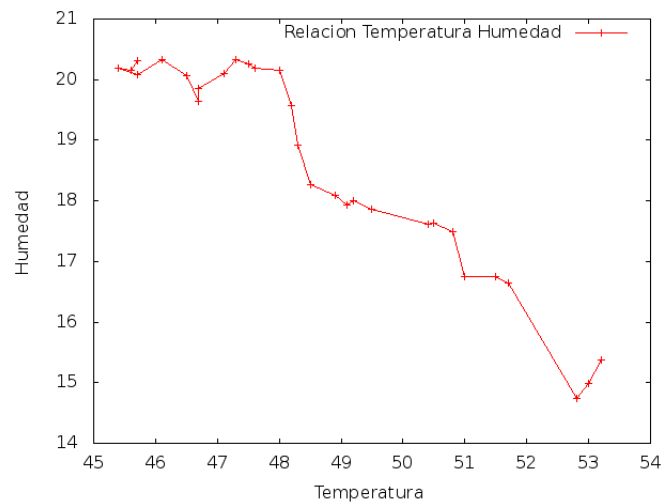


Figura 6.6: Experimento de 12:38 a 12:42 horas

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

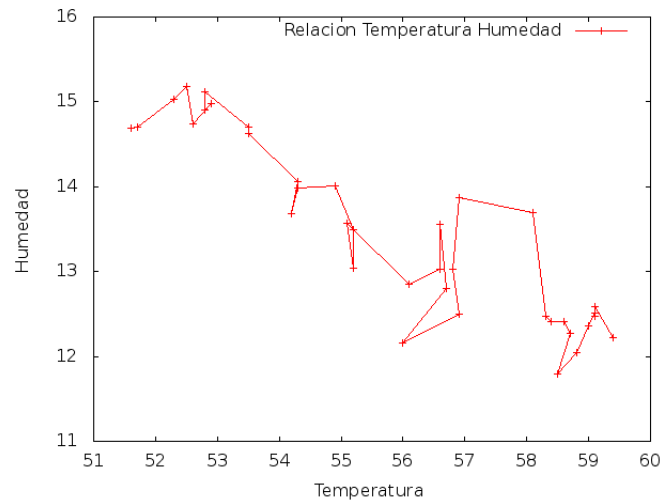


Figura 6.7: Experimento de 14:49 a 14:55 horas

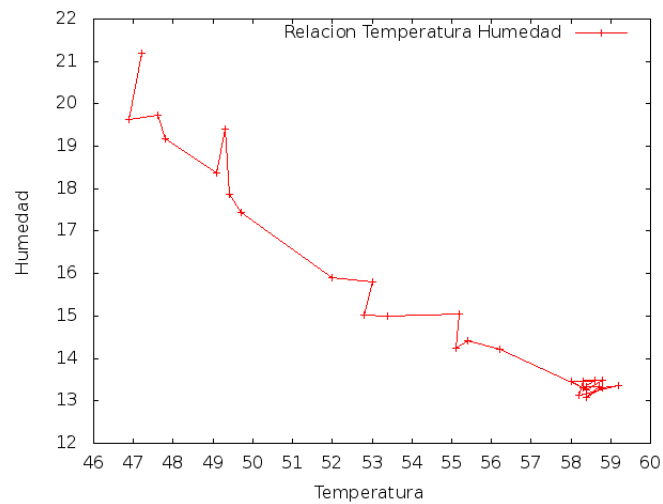


Figura 6.8: Experimento de 15:33 a 15:39 horas



## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

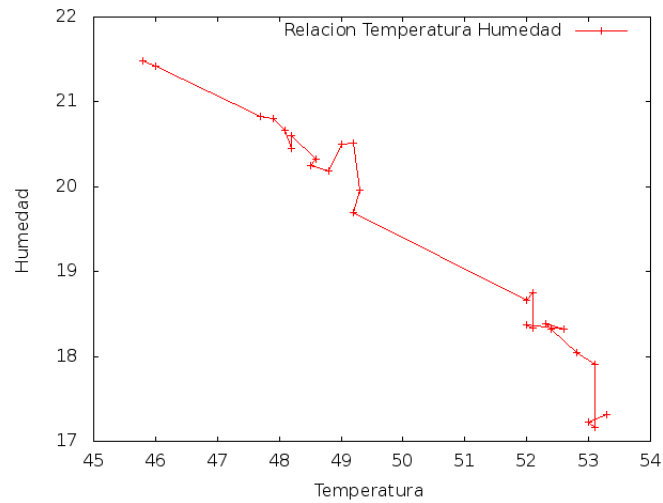


Figura 6.9: Experimento de 17:21 a 17:27 horas

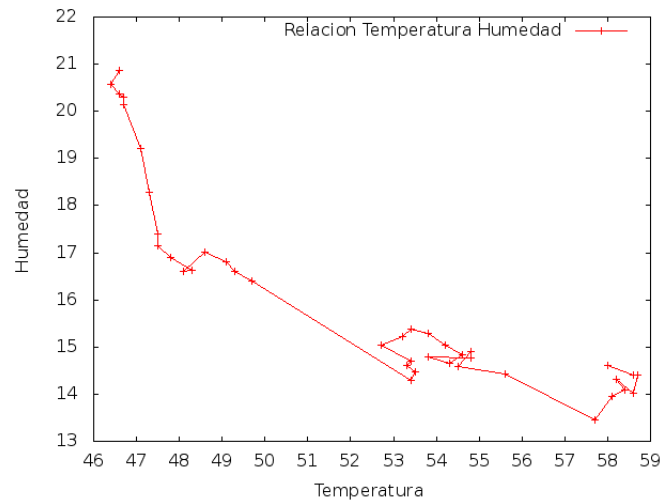


Figura 6.10: Experimento de 17:38 a 17:46 horas

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

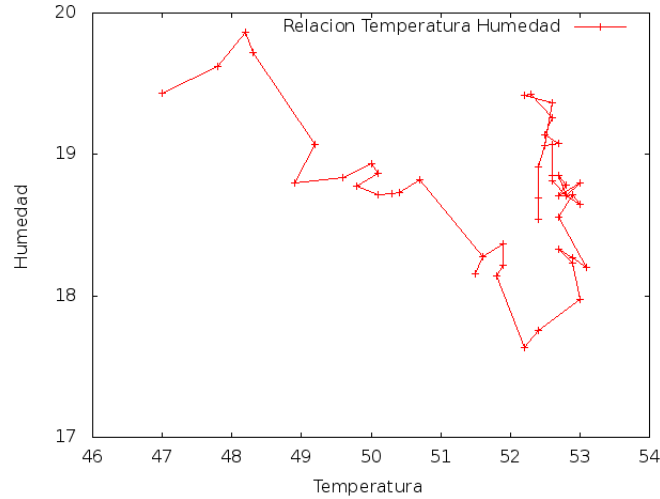


Figura 6.11: Experimento de 18:59 a 19:05 horas

Los datos que se utilizaron para llevar a cabo la interpolación tienen la siguiente gráfica:

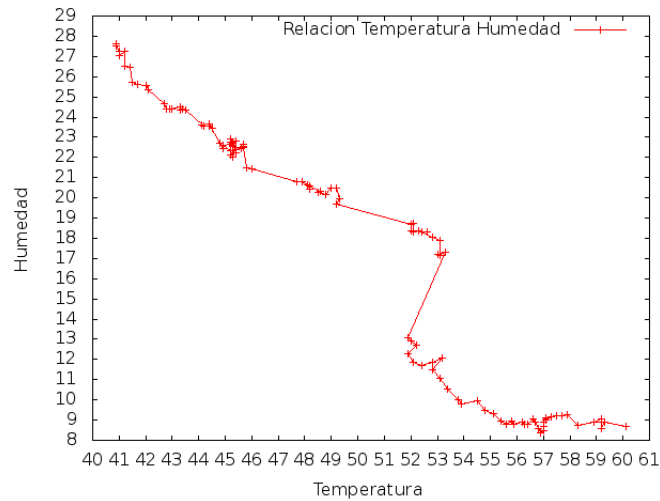


Figura 6.12: Experimentos con fuego

Para llevar a cabo el método utilizando la metodología Dempster-Shafer primero se considera que se supere el umbral de la razón de cambio del promedio de la ventana deslizante de la temperatura el cual desde el algoritmo basado en umbrales fue 1.01, si es sobre pasado se realiza el método, con lo que se obtuvo los siguientes resultados:

## 6.2. MÉTODO PARA DETECCIÓN DE FUEGO BASADO EN DEMPSTER-SHAFER

---

Ventana	Experimentos	Detectados	Falsos Positivos	Falsos Negativos
5	9	6	0	3
6	9	7	0	2
7	9	9	0	0
8	9	9	0	0
9	9	9	0	0
10	9	9	0	0
15	9	9	0	0
20	9	9	0	0
25	9	9	0	0
30	9	9	0	0
35	9	9	0	0

Cuadro 6.2: Resultados del método basado en Dempster-Shafer

En base los resultados obtenidos (ver Cuadro 6.2) de la evaluación del método basado en Dempster-Shafer se obtuvo lo siguiente:

1. Utilizando una ventana deslizante adecuada, para calcular la razón de cambio del promedio de los valores de la temperatura es posible detectar fuego en el ambiente.
2. Se observa una mejoría en comparación a la utilización de la maquina de estados llegando a utilizar una ventana deslizante menor.
3. Los eventos llevados acabo con la presencia de luz fueron detectados utilizando dos sensores, el de temperatura y humedad relativa.
4. En comparación con método basado en umbrales con la maquina de estados, no se presentaron falsos positivos en los eventos llevados acabo en presencia de los rayos del sol.

## Capítulo 7

# Conclusiones y Trabajo Futuro

Las MANETs tienen la capacidad de auto-configurarse utilizando diversas topologías de red, y de comunicarse sin requerir de una infraestructura fija. Las redes de sensores inalámbricas o WSN son un ejemplo de una red MANET. En las WSN, los nodos están equipados con uno o más sensores para medir las condiciones del ambiente. Debido a las limitaciones de recursos con las que cuentan las redes de sensores, es importante utilizar una buena plataforma de software para el desarrollo de aplicaciones.

Las redes de sensores inalámbricas representan una alternativa muy interesante para implementar aplicaciones de detección de eventos. Los incendios forestales causan anualmente una considerable cantidad de pérdidas humanas y materiales cuantiosas, por lo que el desarrollo de sistemas basados en redes de sensores para la detección de incendios tiene una gran importancia.

El método basado en umbrales utiliza los valores de la temperatura, humedad relativa e intensidad de la luz y el método basado en la metodología Dempster-Shafer utiliza los valores de la temperatura y la humedad relativa. Para el diseño y la evaluación de los métodos fue necesario registrar las mediciones en una base de datos durante varios días y generar incendios de manera artificial, llevando un registro de los experimentos. Los resultados obtenidos demuestran que los métodos son capaces de identificar correctamente todos los incendios generados. El método basado en umbrales presentó falsos positivos debido a que los rayos del sol generan cierta similitud con los eventos en presencia de fuego. Es posible solucionar el resultado cubriendo los motes, tratando de simular condiciones con ausencia de luz.

El método basado en Dempster-Shafer no presentó falsos positivos en los experimentos realizados con la presencia de los rayos del sol. Como trabajo futuro, se planea evaluar el método utilizando otros sensores, así como la definición de un nuevo método utilizando análisis numéricos.

# Bibliografía

- [1] Ubuntu home page. <http://www.ubuntu.com/>.
- [2] Ieee 802.15 wpan task group 2 (tg2), Diciembre 2011.
- [3] Tinyos home page. <http://www.tinyos.net/>, Diciembre 2011.
- [4] ZigBee Alliance. Home page zigbee. <http://www.zigbee.org/>, 2011.
- [5] Agustian Taufiq Asyhari. Application of wireless sensor network monitoring system for fire detection in small scale environment. *Seminar Nasional Aplikasi Teknologi Informasi 2006 (SNATI 2006)*, 2006.
- [6] Atmel, 2325 Orchard Parkway San Jose, CA 95131, USA. *IEEE 802.15.4 MAC*, 2006.
- [7] Eric Brewer, David Culler, David Gay, Phil Levis, Rob von Behren, and Matt Welsh. <http://nesc.sourceforge.net/>, nesC Home Page.
- [8] Eric Brewer, David Culler, David Gay, Phil Levis, Rob von Behren, and Matt Welsh. nesc: A programming language for deeply networked systems, December 2004. (Online <http://nesc.sourceforge.net/> Visitado el 13 de Abril de 2011).
- [9] Yong-sork Her Byungrak Son and Jung-Gyu Kim. A design and implementation of forest-fires surveillance system based on wireless sensor networks for south korea mountains. *International Journal of Computer and Network Security*, 6, 2006.
- [10] Imrich Chlamtac, Marco Conti, and Jennifer J. N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, July 2003.
- [11] Hall David L. and Amuya K. Garga. Pitfalls in data fusion (and how to avoid them). *Proceedings of the Second International Conference on Information Fusion (Fusion '99)*, pages 429–436, 1999.
- [12] David M. Doolin and Nicholas Sitar. Wireless sensors for wildfire monitoring.

## BIBLIOGRAFÍA

---

- [13] S. Hadjieftymiades C. Anagnostopoulos E. Zervas, O. Sekkas. Fire detection in the urban rural interface through fusion techniques. 2008.
- [14] Antonio A. F. Loureiro Eduardo F. Nakamura and Alejandro C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computer Surveys*, 39(9):1-55, 2005.
- [15] Frank Cremer Rob Kemp Marco Roos Klammer Schutte Eric Den Breejen, Marcel Breuers and Jan S. De Vries. Autonomous forest fire detection. *In Proceedings of the 3rd International Conference on Forest Fire Research*, 2:2003 – 2012, Noviembre 1998.
- [16] Muhammad Omer Farooq, Sadia Aziz, and Abdul Basit Dogar. State of the art in wireless sensor networks operating systems: A survey. *Lecture Notes in Computer Science*, 6485:616–631, 2010.
- [17] HART Communication Foundation. Tecnologia inalambrica hart, 2011.
- [18] David Gay, Matt Welsh, Philip Levis, Eric Brewer, Robert Von Behren, and David Culler. The nesc language: A holistic approach to networked embedded systems. *In Proceedings of the ACM SIGPLAN conference on Programming language design and implementation*, pages 1–11, 2003.
- [19] Giuseppe C. Marano Giuseppe L. Cascella Giuseppe Giliberti, Giuseppe Lorusso. Hga-based auto-tuning of peltier coolers in pais project: New environmental monitoring and early wildfire detection system. 2009.
- [20] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. *In Proceedings of the 9th international conference on Architectural support for programming languages and operating systems*, pages 93–104, 2000.
- [21] J. A. Gutierrez I. Howitt. Ieee802.15.4 low rate-wireless personal area network coexistence issues. *Wireless Communications and Networking* 3, pages 1481–1486, 2003.
- [22] Weilin Wang Ian F. Akyildiz, Xudong Wang. Wireless mesh networks: a survey. *Elsevier*, 2004.
- [23] ISA. Isa100.11a, release 1. Octubre 2007.
- [24] Biswanath Mukherjee Jennifer Yick and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, pages 2292–2330, Agosto 2008.
- [25] Ning HAN Jiangming KAN Junguo ZHANG, Wenbin LI. Forest fire detection system based on a zigbee wireless sensor network. *Higher Education Press and Springer-Verlang*, 2008.
- [26] Orlando Watkins Kewei Sha, Weisong Shi. Using wireless sensor networks for fire rescue applications: Requeriments and challenges. 2009.

## BIBLIOGRAFÍA

---

- [27] Don Koks and Subhash Challa. An introduction to bayesian and dempster-shafer data fusion. November 2005.
- [28] Marin Vukovic Kresimir Pripuzic, Hrvoje Belani. Early forest fire detection with sensor networks: Sliding window skylines approach. 2008.
- [29] Philip Levis and David Gay. *TinyOS Programming*. Cambridge University Press, 1st edition, April 2009.
- [30] Neng Wang; Xiaoqiao Meng; Liyang Yu. Real-time forest fire detection with wireless sensor networks.
- [31] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 491–502, San Diego, CA, 2003.
- [32] Paul Havinga Majid Bahrepour, Nirvana Meratnia. Automatic fire detection: a survey from wireless sensor network perspective.
- [33] Adi Mallikarjuna, V. Reddy, A.V.U. Phani Kumar, D. Janakiram, and G. Ashok Kumar. Wireless sensor network operating systems: a survey. *International Journal of Sensor Networks*, 5(4):236–255, 2009.
- [34] MEMSIC. Lotus high-performance wireless sensor network platform. 2010.
- [35] MEMSIC. Micaz wireless measurement system. page 3, 2010.
- [36] MEMSIC. Iris wireless measurement system. Noviembre 2011.
- [37] MEMSIC. Telosb mote platform. Noviembre 2011.
- [38] Marco Naeve. Ieee 802.15.4 mac overview. 2004. Eaton Corporation.
- [39] NASA. National environment satellite. <http://modis.gsfc.nasa.gov/>.
- [40] Ryan Newton, Greg Morrisett, and Matt Welsh. The regiment macro-programming system. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 489–498, Cambridge, Massachusetts, 2007.
- [41] Nokia. Home page wibree. <http://www.wibree.com/>, Octubre 2006.
- [42] Evangelos Zervas Odysseas Sekkas. A multi-level data fusion approach for early fire detection. 2010.
- [43] California Department of Forestry and Fire Protection. Fire causes 5-year average 2000-2005.
- [44] Eduardo F. Nakamura Osman S. da Penha Jr. Fusing light and temperature data for fire detection. 2010.

## BIBLIOGRAFÍA

---

- [45] UC Berkeley WEBS Project. Home page nesc. <http://nesc.sourceforge.net/>, Diciembre 2004.
- [46] L. Wang Q. Liang. Event detection in wireless sensor networks using fuzzy logic system. *Computational Intelligence for Homeland Security and Personal Safety*, March 2005.
- [47] M. Franklin S. Madden, R. Szewczyk and D. Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. *IEEE Workshop on Mobile Computing Systems and Applications*, 2002.
- [48] Nukhet SAZAK and Haldun ABDULLAH. The importance of using wireless sensor networks for forest fire sensing and detection in turkey. *5th International Advanced Technologies Symposium (IATS'09)*, Mayo 2009. Turkey.
- [49] Mohammad Shahab. Fault diagnosis: a dempster-shafer theory approach. February 2009.
- [50] Xianyun ZENG Yimin YANG Shaohua CHEN, Hong BAO. A fire detecting method bases on multi-sensor data fusion. 2003.
- [51] Abraham Silberschatz, Peter B. Galvin, and Greg Gagne. *Operating System Concepts*. Wiley, 8th edition, July 2008.
- [52] Ryo Sugihara and Rajesh K. Gupta. Programming models for sensor networks: A survey. *ACM Transactions on Sensor Networks*, 4(2):1550–4859, April 2008.
- [53] N. Markatos V. Vescoukis, T. Olma. Experience from a pilot implementation of an in-situ forest temperature measurement network. *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, 2007.
- [54] H. Vincent Poor Venugopal V. Veeravalli, Tamer Basar. Decentralized sequential detection with a fusion center performing the sequential test. 1993.
- [55] Nazar Elfadil Yaqoob J. AL-Raeisi. Wireless sensor networks performance measurements and monitoring. 2009.
- [56] Jaehyuk Choi Seongho Cho Yeon-sup Lim, Sangsoon Lim. A fire detection and rescue support framework with wireless sensor networks. *2007 International Conference on Convergence Information Technology*, 2007.
- [57] Lei Zhang and Gaofeng Wang. Design and implementation of automatic fire alarm system based on wireless sensor networks. *International Symposium on Information Processing (ISIP'09)*, 2009.