

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Diseño e implementación de las capas, de enlace de datos, y de red para una red inalámbrica de sensores en el Bosque

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Ronny García Ramírez

Cartago, Septiembre de 2006

INSTITUTO TECNOLOGICO DE COSTA RICA

ESCUELA DE INGENIERIA ELECTRONICA

PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Firma

Ing Eduardo Interiano Salguero

Profesor lector

Firma

Ing Faustino Montes de Oca

Profesor lector

Firma

Ing Néstor Hernández Hostatler

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, Costa Rica, Junio 2006

Declaro que el presente Proyecto ha sido realizado por mi persona, utilizando y aplicando literatura referente al tema, así como la información que suministró la institución para la que se realizará el proyecto, y aplicando e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 15 de Junio del 2006



Ronny García Ramírez

Céd: 111370229

Resumen:

El creciente desgaste de los recursos naturales que hacen posible la vida en el planeta plantea la necesidad de desarrollar tecnologías que procuren un mayor control sobre los recursos existentes y lograr de esta forma metodologías de conservación y aprovechamiento ambiental que faciliten el desarrollo sostenible y sostenido de la sociedad. En este sentido la escuela de Ingeniería en electrónica del Instituto Tecnológico de Costa Rica abre una brecha importante en la investigación y desarrollo de tecnologías para la conservación ambiental por medio de una serie de proyectos adjuntos a la vicerrectoría de investigación entre los cuales se encuentra el presente proyecto.

El presente proyecto consiste en el diseño e implementación del protocolo de comunicación utilizado en una red de sensores que permitan el monitoreo de variables diversas en el bosque; el protocolo abarca funciones correspondientes a las capas de red, y de enlace de datos según el modelo de interconexión de sistemas abiertos (OSI)

Entre los principales problemas que se enfrentan en el presente proyecto se encuentran la reducción del consumo energético del módulo de comunicación respecto de un protocolo anteriormente desarrollado, obtener una latencia aceptable en la transmisión de la información, lograr que cada nodo de la red tenga la capacidad de configurarse por si misma de manera que los módulos no tengan que ser modificados en procesos de expansión o falla de alguno de los mismos, y enviar los datos por la ruta más corta posible hacia su destino.

Palabras claves: Red de sensores, consumo energético, latencia, configuración, colisión de tramas, capa de Red, capa de enlace de datos

Summary:

The increasing wearing down of the natural resources that make the life in the planet possible raises the need for developing technologies which performs control of the existing resources and obtains conservation methodologies and environmental advantage that facilitate the sustainable development and sharp of the society. In this sense the school of Electronics Engineering of the Technological Institute of Costa Rica opens the way in the investigation and development of technologies for the environmental conservation by a series of projects which is the present project attached.

The present project consists of the design and implementation of the communication protocol used in a wireless sensor network that allow the monitoring of diverse variables in the forest; the protocol includes functions corresponding to the layers of network, and data link according to the OSI model.

The main problems covered in the present project are the reduction of the power consumption for the communication module respect to a protocol previously developed, to obtain an acceptable latency in the transmission of the information, to obtain that each node of the network has the capacity to auto configure itself in order to obtain modules that doesn't have to be modified in expansion processes or fails of some of such, and to send the data by the possible shortest route towards its destiny.

Key words: Network of sensors, power consumption, latency, configuration, collision, Network layer, data link layer

A mi Dios por darme la fuerza y los recursos para llegar hasta el final, y a mi familia por haber sido sus instrumentos para hacerlo...

Agradezco a los Ingenieros Néstor Hernández, Eduardo Interiano, y Faustino Montes de Oca por sus Valiosos aportes en el desarrollo del presente proyecto.

ÍNDICE GENERAL

Capítulo 1 : Introducción	12
1.1 Problema existente e importancia de su solución	12
1.2 Solución seleccionada	15
Capítulo 2 : Meta y objetivos	16
2.1 Meta:.....	16
2.2 Objetivo general:	16
2.3 Objetivos específicos:	16
a) Objetivos de hardware:	16
b) Objetivos de software	16
c) Objetivos de Documentación	17
d) Objetivos de Implementación.....	17
Capítulo 3 : Marco teórico	18
3.1 Descripción del sistema o proceso a mejorar	18
3.2 Antecedentes bibliográficos.....	18
3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema	20
3.3.1 El modelo de referencia OSI.....	20
3.3.2 Protocolos de la capa de enlace de datos.	22
3.3.3 Asignación estática de canales.....	23
3.3.4 Asignación dinámica de canales.	24
3.3.4 Protocolos de acceso múltiple con detección de portadora	28
3.3.5 Redes de sensores	30
3.3.6 Topologías de Red.....	31
Capítulo 4: Procedimiento metodológico	37
4.1 Reconocimiento y definición del problema.....	37
4.2 Obtención y análisis de información	37
4.3 Evaluación de las alternativas y síntesis de una solución	38
4.4 Implementación de la solución	38
4.5 Reevaluación y rediseño.	39
Capítulo 5 Descripción detallada de la solución	40
5.1 Análisis de soluciones y selección final	40
5.2 Diseño de la solución final	44
5.2.1 Arquitectura de la red:.....	44
5.2.2 Diseño del Protocolo:.....	46
5.2.3 Diseño de las tramas e información de los nodos:.....	53
5.2.3 Diseño de los tiempos del protocolo:	55
5.2.3 Análisis Teórico del protocolo propuesto:	61
5.3 Descripción del hardware	73
5.4 Descripción del Software	74
5.4.1 Estructura General del programa	74
5.4.2: Funciones del modelo OSI implementadas.....	76
Capítulo 6 : Análisis de resultados	79
6.1 Ahorro energético:	79

6.2	Auto configuración:	81
6.3	Confiabilidad:	85
6.4	Escalabilidad:	85
6.5	Independencia:	85
Capítulo 7 : Conclusiones y recomendaciones		86
7.1	Conclusiones	86
7.2	Recomendaciones	86
Bibliografía		88
Apéndices:		89
Apéndice A1: Programa implementado en los nodos		89
Apéndice A2: Antecedentes prácticos:		101
A2.1 Mejoras propuestas al trabajo realizado:		105
Apéndice A3: Suma de variables aleatorias		109

INDICE DE FIGURAS

Figura 2.1	Ilustración del protocolo anteriormente desarrollado.....	14
Figura 4.1:	Ilustración de las capas del modelo OSI	20
Figura 3.2:	Período vulnerable para la trama sombreada	26
Figura 4.3:	Velocidad real de transporte contra tráfico ofrecido en los sistemas aloha.....	27
Figura 3.4:	Comparación de la utilización del canal de los diferentes protocolos....	29
Figura 3.5:	Estados del CSMA/CD	29
Figura 3.6:	Topología en conexión total.....	32
Figura 3.7:	Topología en malla.....	33
Figura 3.8:	Topología en estrella.	34
Figura 3.9:	Topología en estrella extendida.	34
Figura 3.10:	Topología en anillo.....	35
Figura 3.11:	Topología en bus.....	35
Figura 3.12:	Topología en árbol.	36
Figura 5.1:	Ilustración del funcionamiento del protocolo planteado durante la etapa de reconocimiento del problema	40
Figura 5.2:	Ilustración de la arquitectura de la red.....	44
Figura 5.3:	Ilustración del caso en que el alcance del módulo de comunicación es	45
Figura 5.4:	Ilustración de los estados seguidos por cada uno de los nodos del protocolo.....	46
Figura 5.5:	Ilustración de los estados seguidos por el nodo base.....	52
Figura 5.6:	Porcentaje de ahorro en la energía para la red en reposo en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida	57
Figura 5.7:	Ilustración del tiempo de retardo mínimo y máximo que tarda una alarma	63
Figura 5.8:	Ilustración de la diferencia entre el tiempo de retardo mínimo y máximo	63
Figura 5.9:	Función de probabilidad de los tiempos A y B.....	66
Figura 5.10:	Componentes de la convolución de las funciones de densidad de probabilidad a y b.....	67
Figura 5.11:	Función de probabilidad del retraso debido a A+B.....	67
Figura 5.12:	Función de probabilidad del retraso debido a A+B.....	68
Figura 5.13:	Funciones de probabilidad para el retraso para cada una de las	68
Figura 5.14:	Función de probabilidad para el retraso de las tramas en las transiciones entre los nodos de nivel superior a 1 en función de B	68
Figura 5.15:	Diagrama de bloques del hardware para los nodos de la Red.....	73
Figura 5.16:	Diagrama de bloques del hardware para el nodo de base.....	73
Figura 5.17:	Diagrama de flujo de la interrupción del puerto serie	76
Figura 6.1:	Gráfico aproximado del consumo de potencia esperado de la red en reposo utilizando el protocolo propuesto	80
Figura 6.2:	Ilustración del comportamiento obtenido en el proceso de configuración automática.....	83
Figura 6.3:	Cambio en la configuración de los nodos un cambio en la red	84

INDICE DE TABLAS

Tabla 5.1: Descripción de los tipos de alarmas asignadas hasta el momento	54
Tabla 5.2: Porcentaje de ahorro en la energía para la red en reposo, en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida	57
Tabla 5.3: Aumento de la incertidumbre en la estimación del tiempo en el que ocurrió un	64
Tabla 5.4: Porcentaje de probabilidad de que el retardo de transmisión entre 2.....	69
Tabla 6.1: Corriente promedio en los diferentes estados de la etapa de comunicación	79
Tabla 6.2: Porcentaje de ahorro en la energía para la red en reposo, en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida	81

Capítulo 1 : Introducción

1.1 Problema existente e importancia de su solución

El desgaste de los recursos ambientales que hacen posible la vida en el planeta tierra se ha convertido en preocupación de primer orden para la humanidad.

La degradación de los recursos naturales no solamente desemboca en la disminución de las materias primas necesarias en los procesos industriales; además conlleva al deterioro de la calidad de vida de las personas con el agotamiento de los medios necesarios para la subsistencia, tales como: el aire, el agua potable, la protección atmosférica contra las radiaciones solares, el desgaste del suelo para la producción agrícola, entre mucho otros.

Costa Rica a pesar de ser un país pequeño cuenta con un alto porcentaje de la diversidad Biológica del planeta; esto ha convertido a nuestra tierra en un atractivo destino turístico. El turismo en Costa Rica se ha convertido en una floreciente industria que proporciona ingresos significativos a nuestra economía principalmente en zonas rurales donde las oportunidades de desarrollo son escasas, y las fuentes de empleo prácticamente inexistentes.

Es por esto que la conservación ambiental en Costa Rica se convierte en un problema con implicaciones en muchos ámbitos.

En nuestro país a pesar de que existe preocupación por la conservación de los recursos naturales, y que gran parte de nuestro territorio se encuentra protegido por reservas biológicas y parques nacionales no se cuenta con los recursos humanos, económicos, y tecnológicos suficientes que permitan vigilar amplias zonas protegidas. Es por esto que la tala, y la cacería en zonas protegidas por la ley son comunes; y las autoridades carentes de recursos, se ven imposibilitadas para actuar de manera oportuna en pro de la conservación ambiental.

El presente proyecto se enmarca en el desarrollo de una herramienta tecnológica especializada que permita a las autoridades monitorear y proteger de manera efectiva amplias zonas protegidas con un mínimo de personal y recursos económicos de forma que los recursos disponibles se empleen de manera adecuada en las labores más importantes y el personal sea más eficiente.

Específicamente el proyecto consiste en el diseño e implementación del protocolo de comunicación utilizado en una red de sensores que permitan el monitoreo de variables diversas en el bosque; el protocolo debe abarcar funciones correspondientes a las capas de red, y de enlace de datos según el modelo de interconexión de sistemas abiertos (OSI)

Se cuenta con un protocolo anteriormente desarrollado [1] en el que cada uno de los nodos trabaja con comunicación uno a uno por medio de direcciones únicas que identifica cada uno de los módulos de comunicación inalámbrica; en este protocolo los nodos tienen toda la información referente a la topología de la red y las direcciones de sus vecinos configurada de forma manual en sus unidades de procesamiento.

El protocolo de comunicación consiste en que cada vez que uno de los módulos detecta una alarma el mismo pregunta a dos de sus vecinos geográficamente más cercanos al destino de la información por el nivel de carga de batería, una vez que tiene los niveles de carga de las baterías de sus nodos vecinos selecciona el vecino que tiene una carga mayor y envía la información a retransmitir a través de éste. Una vez que éste recibe la información a retransmitir el proceso se repite hasta que la misma llega a su destino.

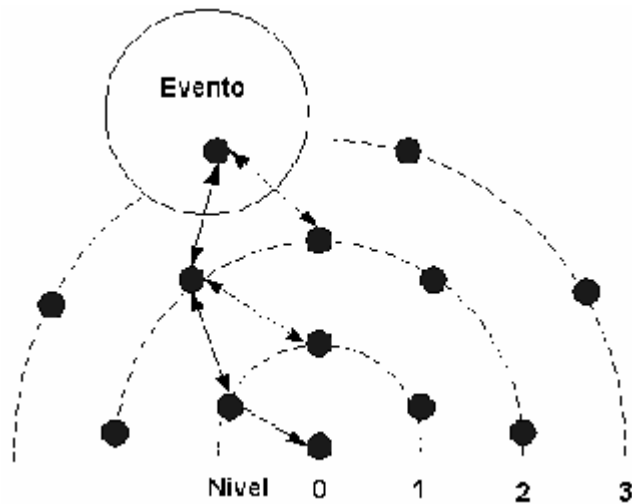


Figura 1.1 Ilustración del protocolo anteriormente desarrollado

Entre los principales problemas que se enfrentan en el presente proyecto se encuentran:

- La reducción del consumo energético del módulo de comunicación respecto del protocolo anteriormente descrito, esto debido a que los módulos de alimentación de energía cuentan con una capacidad limitada de almacenamiento y recarga energética,.
- Lograr que la red sea capaz de configurarse por si misma, de manera que los módulos no tengan que ser modificados en procesos de expansión o falla de alguno de los mismos y además se reduzca la complejidad en la instalación de nuevas redes.
- Eliminar la dependencia del protocolo de los módulos de comunicación que se están utilizando¹, ya que con el prototipo anterior el protocolo se apoya en sus funciones integradas para asegurar la integridad de los datos y la administración del canal de comunicación. Es necesario que el protocolo sea flexible en cuanto a los requerimientos funcionales de la capa física para que pueda migrar fácilmente y ser implementado

¹ Los módulos son los AC4490-1000 de la empresa Aerocomm, puede obtenerse la información técnica de los mismos en la pagina [www. Aerocomm.com](http://www.Aerocomm.com) o en el CD que contiene el presente informe

utilizando dispositivos inalámbricos de comunicación baratos y sencillos.

1.2 Solución seleccionada

La solución seleccionada debe tomar en cuenta los siguientes requisitos:

- Debe mejorar el consumo energético respecto del protocolo anteriormente desarrollado.
- Debe ser flexible para implementarse con la menor cantidad de requerimientos de la capa física, de forma que pueda migrar fácilmente entre diferentes tecnologías.
- Debe ser capaz de configurarse a si misma para simplificar el proceso de instalación y modificación de las redes.
- Debe ser confiable para asegurar que la información llegue correctamente a su destino, o en otras palabras debe contar con algún método de verificación de errores.
- Debe ser capaz de implementarse en los módulos que existen actualmente.

Tomando en cuenta todo lo expuesto se selecciona una solución en la que cada uno de los nodos pasan por una serie de estados predefinidos, lo cual simplifica el proceso de programación, añade modularidad y facilita el proceso de modificación y mejoramiento a demás de los aspectos brevemente descritos en el párrafo anterior.

La solución propuesta se describe en detalle en el capítulo 5 del presente Informe.

Capítulo 2 : Meta y objetivos

2.1 Meta:

Obtener una herramienta de comunicación confiable que permita la transmisión de los datos recolectados en el bosque utilizando un mínimo de recursos energéticos y económicos, de manera que las autoridades tengan conocimiento oportuno de las actividades que dañan la integridad de los ecosistemas y puedan actuar a tiempo en pro de la conservación de la vida silvestre.

2.2 Objetivo general:

Diseñar e implementar un dispositivo de comunicación simple de instalar, confiable, que permita la transmisión de datos recolectados por diferentes teledetectores utilizando los recursos energéticos más eficientemente que el dispositivo anteriormente desarrollado [1].

2.3 Objetivos específicos:

a) Objetivos de hardware:

- El Protocolo desarrollado debe poder ser implementado sobre un Pic16f877a, y un modulo de comunicación inalámbrica AC 4490-100 de la empresa aerocomm.

b) Objetivos de software

- Debe mejorar el consumo energético respecto del protocolo anteriormente desarrollado [1].

- Debe ser flexible para implementarse con la menor cantidad de requerimientos de la capa física, de forma que pueda migrar fácilmente entre diferentes tecnologías.
- Debe ser capaz de configurarse a si misma para simplificar el proceso de instalación y modificación de las redes.
- Debe ser confiable para asegurar que la información llegue correctamente a su destino, o en otras palabras debe contar con algún método de verificación de errores.
- Debe ser capaz de implementarse en los módulos que existen actualmente.

c) Objetivos de Documentación

- Documentar el trabajo realizado de manera detallada de forma que pueda ser continuado, sin complicaciones.

d) Objetivos de Implementación

- Implementar el protocolo sobre los módulos de comunicación existentes.

Capítulo 3 : Marco teórico

3.1 Descripción del sistema o proceso a mejorar

Como se mencionó anteriormente, se cuenta con un protocolo anteriormente desarrollado [1] en el que cada uno de los nodos trabaja con comunicación uno a uno por medio de direcciones únicas que identifica cada uno de los módulos de comunicación inalámbrica; en este protocolo los nodos tienen toda la información referente a la topología de la red y las direcciones de sus vecinos configurada de forma manual en sus unidades de procesamiento.

El protocolo de comunicación consiste en que cada vez que uno de los módulos detecta una alarma el mismo pregunta a dos de sus vecinos geográficamente más cercanos al destino de la información por el nivel de carga de batería, una vez que tiene los niveles de carga de las baterías de sus nodos vecinos selecciona el vecino que tiene una carga mayor y envía la información a retransmitir a través de éste. Una vez que éste recibe la información el proceso se repite hasta que la misma llega a su destino.

3.2 Antecedentes bibliográficos

El interés en las redes de sensores ha crecido substancialmente en los últimos tiempos debido a sus potencialmente importantes aplicaciones en el monitoreo de variables físicas, ubicación de objetos, meteorología, o como en el caso del presente proyecto conservación ambiental [4], [9]. Es por esto que en este momento se realizan una amplia gama de investigaciones que enfocan el crecimiento de este campo en cada área de aplicación específica.

Estas investigaciones han contribuido de manera significativa en la delimitación e identificación de los principales problemas que deben ser abordados en el desarrollo de una red de sensores. Así por ejemplo las investigaciones referidas en [4] y [6] hacen una introducción amplia y valiosa a los objetivos sobre los que debe enfocarse el desarrollo de protocolos sobre

la capa MAC en redes de sensores así como diferentes puntos de vista acerca de los métodos de ahorro energético y administración eficiente del canal en redes con alta densidad de nodos, (dominios de colisión grandes).

En [5] se hace una proposición interesante en el modelado de redes de sensores por medio de un algoritmo de árbol que utiliza información localmente almacenada en cada uno de los nodos de forma que se limita el uso de transmisiones para obtener información de la red, además presenta un enfoque en el que el modelado probabilístico de los sucesos de una red se hace por medios gráficos, facilitando de esta forma la comprensión de los mismos.

En [8] se hace otro tipo de modelado de redes de sensores en el que se hace un análisis profundo e interesante del comportamiento dinámico de la red y de los nodos mientras estos pasan entre estado de actividad y de hibernación. Al final del documento se presentan los resultados obtenidos por medio de la simulación del modelo.

En [7] se discute acerca de la evolución, desarrollo y módulos del hardware utilizado en la construcción de redes de sensores

En [9] se presenta una introducción a los conceptos básicos de las redes de sensores y las redes inalámbricas de computadoras en general.

En [10] se discute acerca de diferentes métodos de enrutamiento que toman en cuenta el ahorro de energía.

3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema

3.3.1 El modelo de referencia OSI.

El modelo de referencia OSI (Modelo de interconexión de sistemas abiertos) es el modelo principalmente utilizado para la descripción de las comunicaciones de red. Lo más importante es que este modelo es una armazón que se puede emplear para comprender cómo viaja la información a través de la red.

El modelo OSI tiene siete capas numeradas, cada una ilustrando una función en particular como se muestra en la figura 3.1



Figura 3.1: Ilustración de las capas del modelo OSI

Esta separación de las funciones de la red se llama división en capas. Dividir la red en estas siete capas proporciona las siguientes ventajas:

- Divide la comunicación de red en partes más pequeñas y sencillas.
- Facilita la normalización de los componentes de la red, al permitir el desarrollo y el soporte de múltiples fabricantes.

- Permiten que diferentes tipos de hardware y software de red se comuniquen entre sí.
- Impide que los cambios de una capa afecten a las otras, por lo que se pueden desarrollar más rápidamente.
- Divide la comunicación de la red en partes más pequeñas para hacer más fácil su comprensión y entendimiento.

Cada capa OSI tiene un conjunto de funciones que debe ejecutar un paquete de datos para viajar desde el origen hasta el destino en una red; a continuación se describen de manera general las funciones de cada una de las capas del modelo [3]:

Capa 7: La capa de aplicación. La capa de aplicación es la capa más cercana al usuario. Proporciona servicios de red, como acceso e impresión de los ficheros para las aplicaciones del usuario. Difiere de las otras capas en que no proporciona servicios a ninguna otra capa OSI sino a las aplicaciones externas al modelo OSI. La capa de aplicación establece la disponibilidad de socios de comunicación deseados. También sincroniza y establece un acuerdo en los procedimientos para la recuperación de errores e integridad en el control de los datos.

Capa 6: La capa de presentación. La capa de presentación asegura que la aplicación que se envía a la capa de aplicación de un sistema se va a poder leer por la capa de aplicación de otro sistema. Si es necesario la capa de presentación traduce múltiples formatos de datos empleando un formato común. Una de las funciones más importantes de esta capa es el cifrado y descifrado.

Capa 5: La capa de sesión. La capa de sesión establece, administra y finaliza las sesiones entre hosts de comunicación.

Capa 4: La capa transporte. La capa de transporte intenta proporcionar un servicio de transporte de datos que proteja a las capas superiores de los detalles de la implementación del transporte. Específicamente lograr un transporte fiable entre dos hosts es el objetivo de la capa de transporte. Al proporcionar un servicio de comunicación, la capa de transporte establece, mantiene y finaliza adecuadamente los circuitos virtuales.

Capa 3: La capa de Red. La capa de red es una capa compleja que proporciona conectividad y una selección de ruta entre dos host que pueden estar ubicados en redes geográficamente separadas. Además la capa de red se ocupa del direccionamiento lógico.

Capa 2: La capa de Enlace de Datos. La capa de enlace de datos proporciona un tránsito de datos fiable a través de un enlace físico. De este modo, la capa de enlace de datos se ocupa del direccionamiento físico de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control de flujo.

Capa 1: La capa física. La capa física define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales.

3.3.2 Protocolos de la capa de enlace de datos.

La capa de enlace de datos se divide en dos subcapas a saber:

- **LLC (Logical Link Control)** Fue creada con el propósito de proporcionar a las capas superiores una interfaz independiente de la tecnología empleada en la capa de enlace de datos y en la capa física.

- **MAC (Media access Control)** Se encarga de la administración del canal de transmisión.

Dado que el presente proyecto se enfoca principalmente a la implementación de la subcapa MAC de la capa de enlace de datos se hará referencia principalmente a los protocolos que a lo largo de la historia se han utilizado para la administración del medio de transmisión.

3.3.3 Asignación estática de canales.

Esta tiene su raíz en la división que tradicionalmente se hace de una troncal telefónica entre varios usuarios por medio de FDM. Esta técnica tiene sentido cuando existe un número pequeño y constante de usuarios que tienen los suficientes datos para mantener ocupado el ancho de banda que se le asigne; no obstante existe desperdicio del ancho de banda cuando algunos usuarios no transmiten o transmiten por ráfagas.

El desempeño pobre de la FDM estática se puede ver fácilmente por medio de un cálculo sencillo de la teoría de colas. Si se tiene un medio con un retardo medio T , de un canal de C bps de capacidad con una tasa de llegada de λ tramas/segundo, en el que cada trama tiene una longitud que se obtiene de una función exponencial de densidad de probabilidad con una media de $1/\mu$ bits/trama. Con estos parámetros la tasa de llegada es de λ tramas/segundo y la tasa de servicio es de μC tramas/segundo. A partir de la teoría de colas se puede demostrar que para tiempos de llegada y de servicio de Poisson,

$$T = \frac{1}{\mu C - \lambda} \quad (3.1)$$

Si se subdivide el canal en N subcanales independientes cada uno con capacidad de C/N bps la tasa media de entrada de cada uno de los subcanales ahora será de λ/N . Recalculando T, obtenemos:

$$T_{FDM} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT \quad (3.2)$$

El retardo utilizando FDM es N veces peor que si todas las tramas se acomodaran en una gran cola central.

Los mismos argumentos planteados anteriormente para FDM son válidos para TDM.

3.3.4 Asignación dinámica de canales.

Aloha

Este método fue desarrollado en la década de los 70's en la Universidad de Hawai usando la radiodifusión. Existen dos variantes de este método que son el Aloha puro y el ranurado.

Aloha puro:

La idea básica es permitir que los usuarios transmitan cuando tengan datos por enviar. Por supuesto habrá colisiones y las tramas en colisión se dañarán. Sin embargo por la propiedad de retroalimentación de la difusión un emisor siempre podrá saber si la trama fue destruida o no escuchando el canal. Si existe una colisión el emisor espera un tiempo aleatorio e intenta de nuevo; el tiempo debe ser aleatorio o de lo contrario las tramas chocarán una y otra vez.

Para analizar la eficiencia del aloha puro se supondrá una cantidad infinita de usuarios que genera tramas nuevas según una distribución de poisson con media de N tramas por tiempo de trama (La suposición de usuarios infinitos es necesaria para asegurar que N no disminuya a medida que se bloquean los usuarios) Si $N > 1$ la comunidad de usuarios está generando tramas a una velocidad mayor que la que puede manejar el canal y casi cada trama sufre una colisión. Para una velocidad real de transporte razonable esperaríamos que $0 < N < 1$.

Además de tramas nuevas las estaciones también generan retransmisiones de tramas que previamente sufrieron colisiones. Asimismo supongamos que la probabilidad de K intentos de transmisión por tiempo de trama, viejas y nuevas combinadas, también es una distribución de poisson con una media de G por intento de trama. Claramente $G \geq N$. Con todas las cargas la velocidad real de transporte S es solo la carga ofrecida, G por la probabilidad P_0 de que una transmisión tenga éxito (Es decir $S = GP_0$ donde P_0 es la probabilidad de que una trama no sufra una colisión). Una trama no sufrirá una colisión si no se envía otras tramas durante un tiempo de trama desde su envío. Como se muestra en la figura 3.2

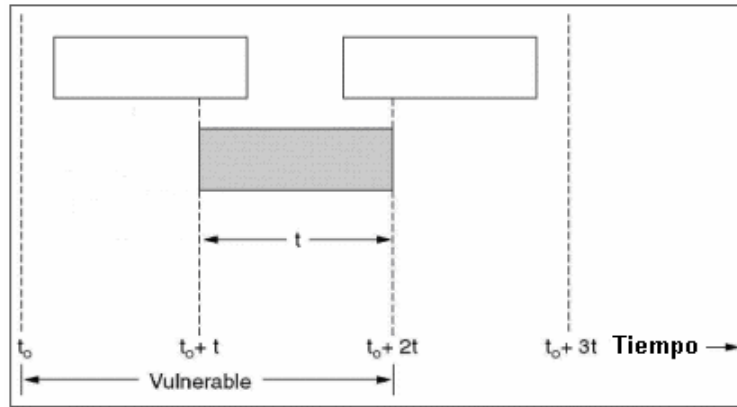


Figura 3.2: Período vulnerable para la trama sombreada²

La probabilidad de que K tramas sean generadas durante un tiempo de trama determinado está dada por la distribución de Poisson:

$$\Pr[k] = \frac{G^k e^{-G}}{K!} \quad (3.3)$$

Así que la probabilidad de cero tramas es simplemente e^{-G} . En un intervalo de dos tiempos de trama de longitud, el número medio de tramas generadas es $2G$. La probabilidad de que no se inicie otro tráfico durante todo el periodo vulnerable está dado entonces por $P_0 = e^{-2G}$. Si $S = GP_0$ Se obtiene que:

$$S = Ge^{-2G} \quad (3.4)$$

En la figura 4.4 se muestra la relación entre el tráfico ofrecido y la velocidad real de transporte. La velocidad de transporte máxima ocurre a $G = 0.5$, con $S = 1/2e$, que es aproximadamente 0.184, en otras palabras lo más que podemos esperar es un uso del canal del 18%

Aloha ranurado:

Es una mejora que se realiza sobre el aloha puro y consiste en dividir el tiempo en intervalos discretos, cada uno de los cuales corresponde a una trama. Este enfoque requiere que los usuarios acuerden límites de ranura. Una manera de lograr la sincronización sería tener una estación

² Figura tomada de [2]

especial que emitiera una señal al comienzo de cada intervalo como un reloj. A diferencia del aloha puro no se permite que una estación transmita cada vez que tiene datos en la cola. En cambio se le obliga a esperar a la siguiente ranura.

Dado que el período vulnerable es ahora de la mitad la probabilidad de que no haya más tráfico durante la misma ranura que nuestra trama de prueba es de e^{-G} , lo que conduce a:

$$S = Ge^{-G} \quad (3.5)$$

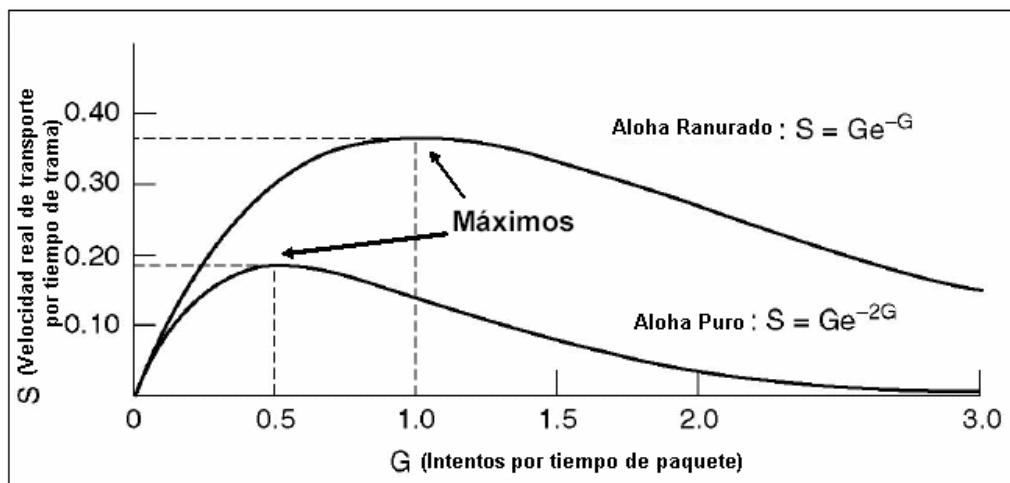


Figura 3.3: Velocidad real de transporte contra tráfico ofrecido en los sistemas aloha³

Como se puede ver en la figura 3.3 el aloha ranurado alcanza su máximo valor en $G=1$, con una velocidad real de transporte de $S=1/e$, o aproximadamente 0.368, el doble de aloha puro. Si el sistema está operando a $G=1$ lo mejor que podemos esperar usando aloha ranurado es 37% de ranuras vacías, 37% de éxitos, y 26% de colisiones.

³ Figura tomada de [2]

3.3.4 Protocolos de acceso múltiple con detección de portadora

CSMA:

El CSMA (Acceso múltiple con detección de portadora) es un protocolo en el que cuando una estación tiene datos para transmitir primero escucha el canal para saber si otra estación está transmitiendo en ese momento. Si el canal está ocupado la estación espera hasta que el canal se desocupe. Cuando la estación detecta un canal inactivo transmite la trama; si ocurre una colisión la estación espera una cantidad aleatoria de tiempo y comienza de nuevo.

Existen diferentes variantes de este protocolo a saber:

CSMA 1-persistente: El protocolo inicia transmisión con una probabilidad 1 cuando encuentra el canal libre después de esperar

CSMA no persistente: Al encontrar el canal ocupado, no se monitorea éste para ocuparlo inmediatamente que se encuentre libre; sino, que se espera un tiempo aleatorio y luego se prueba

CSMA p-persistente: En canales de tiempo discreto, el protocolo inicia la transmisión con una probabilidad p cuando encuentra el canal libre después de esperar o la difiere con $q = (1-p)$.

En la figura 3.4 se muestra una comparación entre los diferentes protocolos

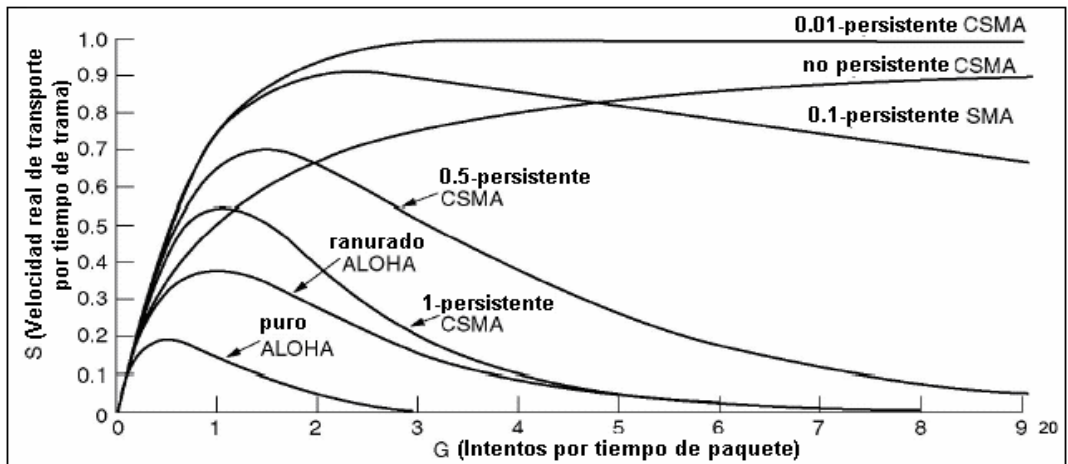


Figura 3.4: Comparación de la utilización del canal de los diferentes protocolos⁴

CSMA / CD:

El CSMA con detección de portadora funciona básicamente con el mismo principio que el CSMA solamente que cuando una estación detecta una colisión, aborta la transmisión espera un tiempo aleatorio e intenta de nuevo, suponiendo que ninguna estación ha comenzado a transmitir durante ese lapso. El modelo consistirá de contención y transmisión, ocurriendo periodos de inactividad cuando todas las estaciones se encuentren en reposo. Esta interacción se representa en la figura 3.5.

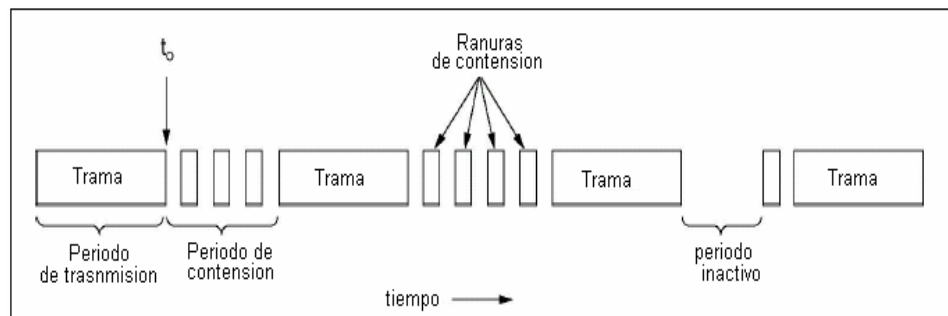


Figura 3.5: Estados del CSMA/CD⁵

⁴ Figura tomada de [2]

⁵ Figura tomada de [2]

3.3.5 Redes de sensores

Una red de sensores es una red inalámbrica compuesta de sensores pequeños y de bajo costo, los cuales se encargan de recolectar información del entorno. Este tipo de redes facilitan la vigilancia y el control del entorno físico desde ubicaciones remotas con una mayor precisión. Entre sus aplicaciones se pueden mencionar la vigilancia ambiental, asuntos militares y la recolección de información de lugares de difícil acceso.

Los nodos de sensores tienen limitaciones tanto de consumo de potencia y de capacidad de procesamiento debido a su bajo costo y infraestructura del entorno donde es implementado el sistema. Para contrastar estas deficiencias se hace uso de más sistemas más eficientes de enrutamiento que disminuyen el consumo de potencia, algoritmos de localización de nodos y diseño planificado del sistema. El uso de varios sensores se justifica cuando se desea realizar un estudio de un fenómeno que no tiene una ubicación conocida, permitiendo un análisis más amplio. Además, en muchos casos, múltiples sensores son requeridos para eliminar obstrucciones y limitaciones de línea vista, entre otras.

A pesar de las diversas aplicaciones de este tipo de red, existen varios obstáculos que se deben superar debido a los siguientes factores:

Independencia de conexiones inalámbricas

La mayoría de los nodos de sensores son ubicados en regiones donde no existe ningún tipo de infraestructura. Ejemplo de ello es cuando se desea realizar una red de este tipo en un bosque. En este caso la identificación de la conectividad y distribución de información depende de la capacidad de los nodos.

Inalteración

Los nodos de sensores no están conectados a ninguna infraestructura que logre suplir el consumo de energía. Solo existe una fuente finita de energía que debe ser optimizada para el procesamiento de datos y la comunicación entre nodos. El mayor consumo de energía se debe al proceso de comunicación, por lo que se debe de minimizar en la medida posible.

Cambios dinámicos

Es necesario que las redes de sensores sean adaptables a cualquier cambio en la conectividad (por la adición de más nodos, falla en algún nodo, etc.), Además de cambios en el entorno mismo.

3.3.6 Topologías de Red

La disposición de los diferentes componentes de una red se conoce con el nombre de topología de red. Existen dos tipos de topologías en una red de comunicaciones:

- *Topología Física:* comprende la disposición real de los dispositivos de comunicación.
- *Topología Lógica:* describe el comportamiento del flujo de información a través de una red, lo cual es utilizado para determinar donde pueden producirse las colisiones.

Una red puede tener un tipo de topología física y un tipo completamente diferente de topología lógica.

Una red de comunicaciones está compuesta de nodos, los cuales, además de su capacidad de procesamiento de datos, tienen como característica la transmisión y recepción de datos a través de los enlaces establecidos, ya sea por un medio cableado o inalámbrico. Los dos tipos más comunes de topologías lógicas son Broadcast y transmisión de tokens.

Topología Broadcast

En esta configuración cada nodo envía sus datos hacia todos los demás nodos sin considerar un orden específico. Los dispositivos no siguen ningún orden para utilizar la red, sino que cada uno accede a la red para transmitir datos en el momento en que lo necesite.

Topología Token

La transmisión de tokens controla el acceso a la red al transmitir un token eléctrico de forma secuencial a cada nodo. Cuando un nodo recibe un token, significa que puede enviar datos a través de la red. Si el nodo no tiene ningún dato para enviar, transmite el token hacia el siguiente nodo y el proceso se vuelve a repetir.

Entre las topologías físicas más utilizadas se pueden mencionar las siguientes:

Topología en conexión total (malla completa)

Su principal característica es que todos los nodos están entre conectados, proporcionando un sistema robusto, como se muestra en la figura 3.6. En el caso de que algún enlace falle, los datos son transmitidos por una nueva ruta hacia su destino. Esta configuración es empleada en los casos en que una ruptura en la comunicación no es posible.

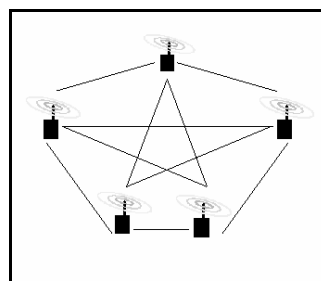


Figura 3.6: Topología en conexión total

Este tipo de topología presenta el problema de que conforme se aumenta la cantidad de nodos, la cantidad de enlaces se incrementa exponencialmente, aumentando la complejidad de la misma. Por lo tanto, en redes amplias, se presenta un problema de enrutamiento aún cuando se cuente con recursos de gran poder de procesamiento computacional.

Topología en malla

Esta topología tiene distribuidas redes en forma regular que generalmente solo pueden comunicarse con el vecino más cercano del nodo, como se muestra en la figura 3.7. Normalmente los nodos de estas redes son idénticos, debido a ello, las mallas también son conocidas como redes de igual a igual (peer to peer). Debido a que existen múltiples caminos de enrutamiento entre nodos, este tipo de redes son robustas ante fallas individuales de nodos o enlaces. Una ventaja de la conexión malla es que, al ser todos los nodos idénticos y tener el mismo nivel de procesamiento y transmisión, ciertos nodos pueden ser escogidos como “líderes de grupo”, los cuales realizan funciones adicionales. Si un líder de grupo es deshabilitado, entonces otro nodo puede tomar sus funciones.

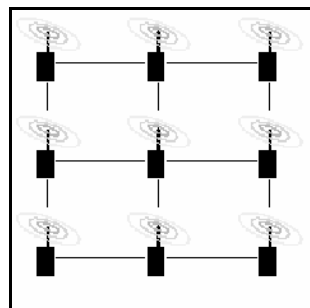


Figura 3.7: Topología en malla

Topología en estrella

En esta configuración todos los nodos están conectados a un solo nodo concentrador, la figura 3.8 ilustra esto. Dicho concentrador manipula una mayor cantidad de datos, procesos de enrutamiento y capacidad de toma de decisiones que los otros nodos. Si uno de los enlaces es deshabilitado, solo afecta un solo nodo. Sin embargo, si el concentrador deja de funcionar toda la red desaparece.

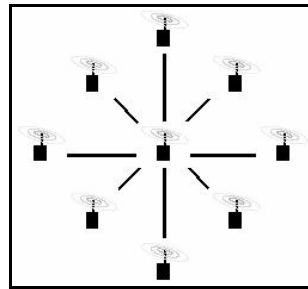


Figura 3.8: Topología en estrella.

Topología en estrella extendida

Este tipo de configuración emplea la topología estrella como base, su diagrama se muestra en la figura 3.9. Esta enlaza diversas estrellas a través de sus nodos concentradores, extendiendo de esta manera la longitud y el tamaño de la red.

Presenta como ventaja que limita el número de dispositivos necesarios para interconectar cualquier nodo concentrador.

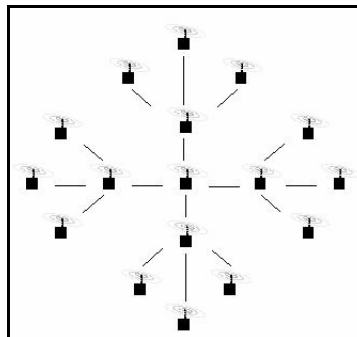


Figura 3.9: Topología en estrella extendida.

Topología en anillo

En esta topología todos los nodos realizan las mismas funciones y no existe un nodo líder, esto se observa en la figura 3.10. Los datos por lo general viajan en una sola dirección. Sin embargo si el anillo es cortado, entonces toda la comunicación se pierde. Para evitar el caso antes mencionado se emplea una configuración de doble anillo, utilizando un anillo primario para el funcionamiento normal y uno de respaldo en caso de que el primario sea cortado.

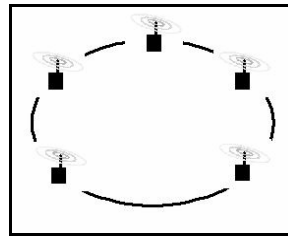


Figura 3.10: Topología en anillo.

Topología en bus

Esta configuración tiene todos los nodos directamente conectados a un solo enlace (backbone), los datos son enviados a través del bus, como lo muestra la figura 3.11. Cada uno de los nodos verifica la dirección destino en el encabezado del mensaje, en caso de que no corresponda con su dirección, el mensaje es descartado. Este tipo de topología es pasiva en el sentido de que cada nodo simplemente escucha el mensaje y no es responsable de la retransmisión de ningún mensaje. Como ventaja se puede mencionar que todos los dispositivos están conectados entre sí, en los casos en que se desee una comunicación directa. Por otro lado, si se presenta una ruptura del enlace, se pierde la comunicación entre dispositivos.

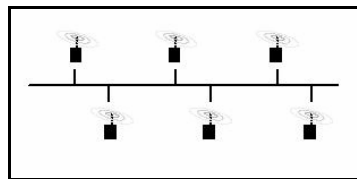


Figura 3.11: Topología en bus.

Topología en árbol

Esta topología es similar a la estrella extendida. Su diferencia radica en que no se utiliza un nodo concentrador, sino que utiliza un nodo troncal desde el que se ramifica a otros nodos, como se observa en la figura 3.12. El flujo de información es jerárquico.

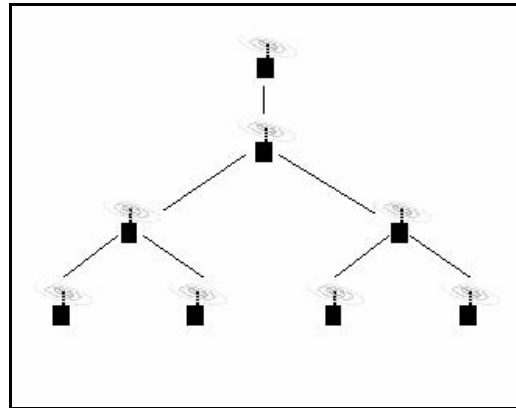


Figura 3.12: Topología en árbol.

La principal consideración al implementar una red de comunicación es la calidad de servicio (QoS) que se quiere ofrecer. Entre los factores que afectan la calidad de procesamiento de datos en una red se pueden mencionar los siguientes:

- Retrasos en la transmisión de datos
- Tasa de error por bit, pérdidas de paquetes de información
- Costo económico de las transmisión
- Consumo de potencia

Dependiendo de la calidad de servicio, consideraciones económicas, el entorno donde se instale la red y la aplicación, se determina el tipo de topología a utilizar. Una red puede estar constituida de varias subredes interconectadas con diferentes tipos de topologías.

Capítulo 4: Procedimiento metodológico

4.1 Reconocimiento y definición del problema

Se resalta en este punto la valiosa contribución prestada por el ingeniero a cargo del proyecto: profesor Ing Néstor Hernández, y los profesores Ingenieros Faustino Montes de Oca, y Eduardo Interiano en el proceso de definición de las prioridades y necesidades principales en el desarrollo del proyecto. Además se realizó una entrevista telefónica con el Ing Emmanuel Navarro, el cual trabajó anteriormente en el proyecto y con su perspectiva ayudó a aclarar el panorama del mismo. A esto se une la realización de pruebas⁶ sobre el trabajo realizado anteriormente por Emmanuel.

Partiendo de la información obtenida y la aplicación de los conocimientos alcanzados a lo largo de la carrera y especialmente en el curso de Redes de computadoras se definen las metas y restricciones generales establecidas para la solución que se propone y explica con detalle en el capítulo 5 del presente informe.

4.2 Obtención y análisis de información

Para la obtención de la información relevante utilizada para el planteamiento y realización del presente proyecto se hizo uso especialmente de la bibliografía disponible y referenciada en la Bibliografía además del análisis de trabajos previamente realizados en el campo de las redes inalámbricas y las redes de sensores.

El análisis de la información obtenida y posterior diseño de la solución implementada se hizo con base en los requerimientos del problema

⁶ En el apéndice a9 se muestran algunas de las pruebas realizadas

anteriormente descritos y las capacidades de implementación con las que se cuentan en el proyecto.

4.3 Evaluación de las alternativas y síntesis de una solución

La solución propuesta en el capítulo 5 del presente informe es el resultado de la modificación de una propuesta más simple hecha a partir de la investigación bibliográfica realizada así como el análisis de los requerimientos del proyecto y modificada con base en el análisis teórico del comportamiento del protocolo principalmente en aspectos como la latencia, la certeza de que la trama de alarma llegue sin errores a su destino, y la escalabilidad de la red. Además la propuesta se ha evaluado en función del comportamiento mostrado en diferentes pruebas realizadas en la implementación del mismo.

Es importante destacar en este punto que a pesar de que existen referencias abundantes de proyectos realizados en el área de las redes de sensores los cuales han servido como guía en el proceso de exploración de diferentes posibilidades de implementación, ninguna se adapta exactamente a los requerimientos que se plantean en el presente proyecto.

4.4 Implementación de la solución

En el proceso de implementación de la solución, como se comento en el apartado anterior, se notaron algunas deficiencias del planteamiento hecho hasta ese momento, principalmente en la temporización de las tramas, las cuales se corrigieron en esta etapa; además se hicieron algunas modificaciones en el protocolo propuesto para la estación base con el objetivo de evitar la modificación física del nodo existente para este propósito.

Para analizar el comportamiento del protocolo fue necesaria la creación de pruebas en las que se simularon diferentes ambientes para aprobar la

capacidad que tienen los nodos de configurarse por sí mismos ante cambios en la red, la latencia en la transmisión de la información, y la colisión de tramas, estas pruebas se hicieron usando Matlab.

Para dar a conocer el trabajo realizado se realizó una entrevista en la que se explico detalladamente todas las acciones realizadas al asesor del proyecto. Además se entrega el presente informe y un disco con toda la información relevante del software diseñado.

4.5 Reevaluación y rediseño.

Es importante mencionar que en el momento en el que se realizó el presente proyecto no se contaba con los sensores para hacer pruebas con el módulo completo, es por esto que todo el desarrollo se realizo por medio del uso de microcontroladores del tipo PIC 16F877, para futuros procesos de reevaluación y rediseño además de tomar en cuenta los parámetros básicos de evaluación del protocolo como lo son la latencia, la administración del canal, y el consumo de energía, desarrollar la aplicación en un dispositivo que permita la integración de todas las partes del proyecto como por ejemplo una FPGA.

Actualmente existe un grupo de trabajo del IEEE (Institute of Electrical and Electronics Engineers) que se encuentra desarrollando el IEEE 1451 Sensor Networks, el cual es un estándar aplicable estrictamente para redes inalámbricas de sensores, en el que se especificarán los detalles para el diseño y construcción de este tipo de sistemas y deberá ser tomado en cuenta cuando se llegue a integrar el trabajo de los diferentes frentes de trabajo involucrados en la consecución de la meta de este proyecto.

Capítulo 5 Descripción detallada de la solución

5.1 Análisis de soluciones y selección final

Inicialmente durante el proceso de reconocimiento del problema se establece evitar el uso de faros para la administración del canal debido a que supone una mayor complejidad en la instalación de los sistemas.

Es por esto que se planteó un protocolo cuya estrategia es que el emisor estimule al receptor a enviar una trama corta de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas hacerlo durante la siguiente trama de datos⁷.

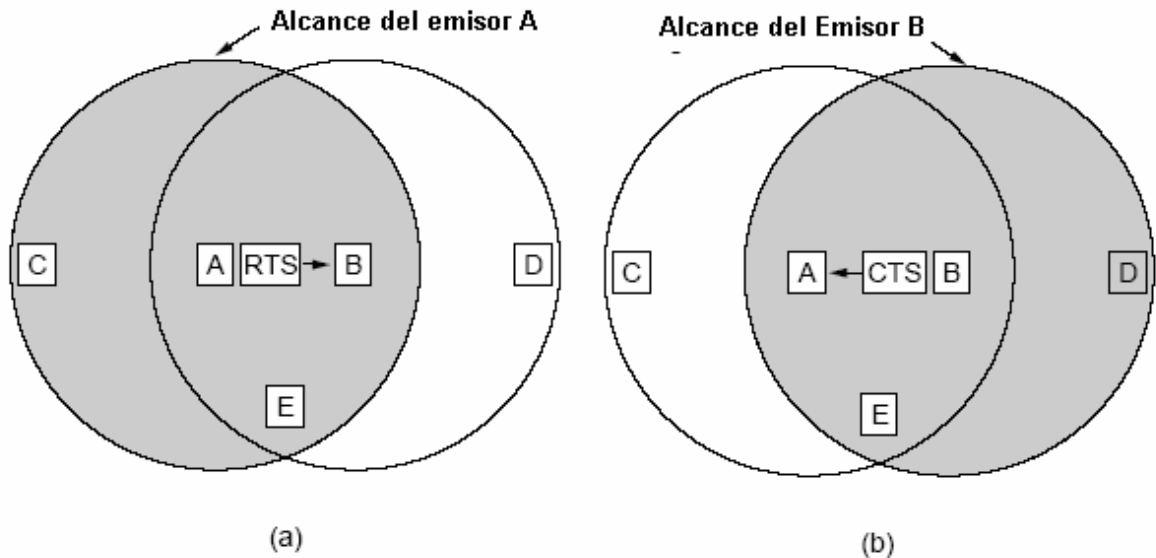


Figura 5.1: Ilustración del funcionamiento del protocolo planteado durante la etapa de reconocimiento del problema⁸

⁷ Este protocolo es el que se conoce como MACAW (Acceso múltiple con prevención de colisiones inalámbrico), y fue desarrollado inicialmente por Karn, Bharghavan, y Cols entre 1990 y 1994. [2]

⁸ Figura tomada de [2]

En la figura 5.1 se ilustra el concepto. Considere la manera en que A envía una trama a B. A comienza por enviar una trama RTS (solicitud de envío) a B, como se muestra en la figura 1(a), esta trama contiene la longitud de la trama que seguirá posteriormente⁹. Después B contesta con una trama CTS (Libre para el envío) como se muestra en la figura 1(b) la trama CTS contiene la longitud de los datos copiada de la trama RTS. Una vez que sucede la recepción de la trama CTS, A comienza a transmitir.

Cualquier estación que escuche el RTS evidentemente está bastante cerca de A y debe permanecer en silencio el tiempo suficiente para que el CTS se transmita de regreso a A sin conflicto.

Cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS.

En la figura 5.1, C está en el alcance de A, pero no en el de B. Por lo tanto escucha el RTS de A pero no el CTS de B. Mientras no interfiera con el CTS está libre para transmitir mientras se está enviando la trama de datos. En contraste, D está en el alcance de B pero no de A. No escucha el RTS pero si el CTS. Al escuchar el CTS se le indica que esta cerca de una estación que esta a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta el momento en que espera la terminación de esta trama. La estación E escucha ambos mensajes de control y, al igual que D, debe permanecer en silencio hasta que halla completado la trama de datos.

A pesar de estas precauciones, aún pueden ocurrir colisiones. Por ejemplo, B y C pueden enviar tramas RTS a A. Al mismo tiempo estas chocarán y se perderán. En el caso de una colisión, un emisor sin éxito (es decir uno que escucha un CTS) en el intervalo de tiempo esperado) espera un tiempo aleatorio y reintenta. El algoritmo que se plantea para este efecto es el retroceso exponencial binario.

⁹ Se puede hacer la trama más corta si se hace que esta longitud tenga un valor fijo.

El algoritmo de retroceso exponencial binario es utilizado en redes del tipo ethernet para calcular el tiempo que debe esperar una estación antes de reintentar la toma del canal una vez que ha ocurrido una colisión (o sea que dos tramas intenten tomar el canal al mismo tiempo y se destruyan).

Tras una colisión el tiempo se divide en ranuras discretas cuya longitud es igual al tiempo máximo que dura en transmitirse de manera exitosa la trama más larga de una estación a otra.

Tras la primera colisión la estación espera 0 ó 1 tiempo de ranura antes de intentarlo de nuevo. Si dos estaciones entran en colisión y ambas escogen el mismo número aleatorio, habrá una nueva colisión. Después de la segunda colisión cada una escoge 0, 1, 2, ó 3 al azar y espera ese número de tiempos de ranura, si ocurre una tercera colisión (la posibilidad de que esto ocurra es de 0.25), entonces para la siguiente vez el número de ranuras a esperar se escogerá al azar del intervalo de 0 a $2^3 - 1$.

En general, tras i colisiones, se escoge un número aleatorio entre 0 y $2^i - 1$, se salta ese número de ranuras, sin embargo tras haberse alcanzado 10 colisiones, el intervalo de aleatorización se congela en un máximo de 1023 ranuras. Tras 16 colisiones el controlador colapsa e informa al computador de una falla. La recuperación posterior es responsabilidad de capas superiores.

Este algoritmo se escogió para adaptar en forma dinámica el número de estaciones que intentan transmitir. Si el intervalo de aleatorización para todas las colisiones fuera de 1023 ranuras, la posibilidad de que chocaran dos estaciones una segunda vez sería insignificante, pero la espera promedio tras una colisión será de cientos de tiempos de ranura, lo introduce un retardo significativo. Por otra parte si cada estación siempre se retarda 0 ó 1 ranuras, entonces al tratar de transmitir 100 estaciones al mismo tiempo, habría colisiones una y otra vez hasta que 99 de ellas escogiera 1 y la estación restante escogiera 0. Esto podría tomar años. Haciendo que el intervalo de aleatorización crezca de manera exponencial a

medida que ocurren más y más colisiones, el algoritmo asegura un retardo pequeño cuando solo unas cuantas estaciones entran en colisión, pero también asegura que la colisión se resuelva en un intervalo razonable cuando hay colisiones entre muchas estaciones. Truncar el proceso a 1023 evita que el límite crezca demasiado.

De lo anterior se puede notar que sin confirmación de recepción de la capa de enlace de datos las tramas no son retransmitidas en caso de que se transmitan con errores. Este problema planteaba resolver introduciendo una trama ACK (reconocimiento) tras cada trama de datos transmitida exitosamente.

A pesar de que posteriormente se siguen implementando las tramas de RTS y CTS para administrar el canal, y se mantiene la idea inicial de utilizar tramas de ACK para asegurar la fiabilidad de los datos el protocolo anteriormente expuesto utiliza más tramas de control que el protocolo inicial [1], y mantiene al igual que el anterior todos los nodos de la red permanentemente encendidos por lo que simplemente supone un consumo mayor de energía. Es por esto que se desecha esta idea y se busca un método en el que los nodos puedan ahorrar energía respecto del modelo anterior [1].

5.2 Diseño de la solución final

5.2.1 Arquitectura de la red:

La arquitectura de la red es aproximadamente una estrella extendida, en la que la conexión entre los nodos se encuentra condicionada al alcance de la etapa de radiofrecuencia; como se muestra en la figura 5.2:

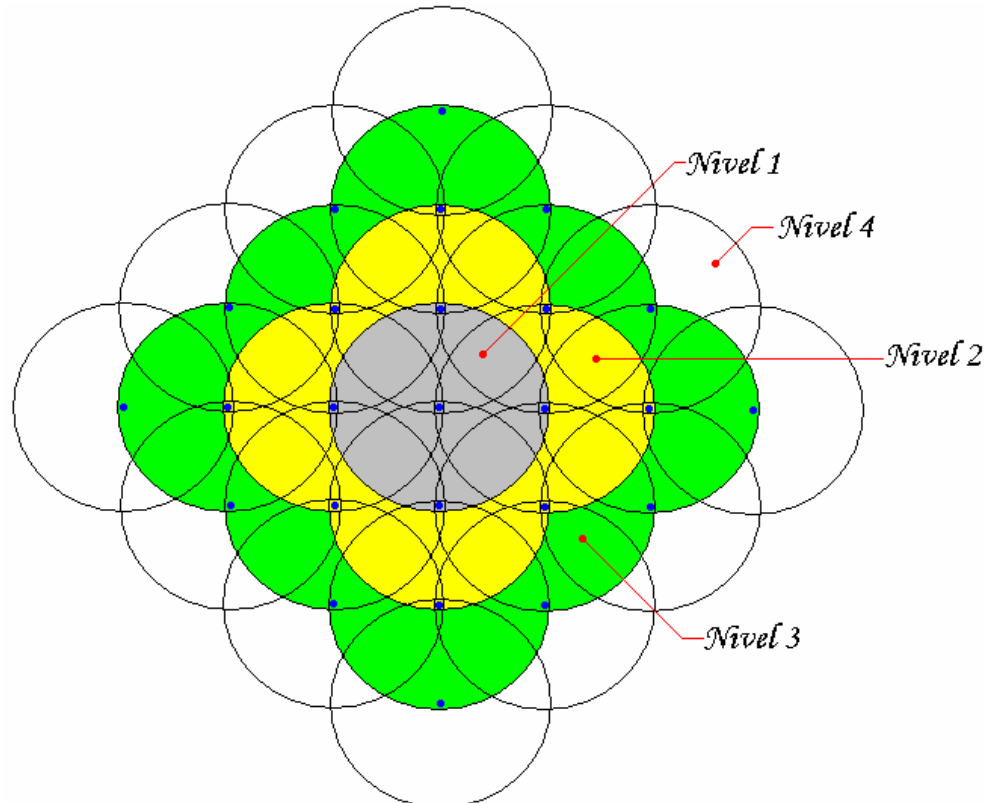


Figura 5.2: Ilustración de la arquitectura de la red

Partiendo de esto se puede definir una red organizada jerárquicamente en niveles, en la que pueden definir como nodos de nivel 1 todos los nodos que se encuentren en el alcance de la estación base (zona gris), los nodos de nivel 2 serán los que se encuentren en el alcance de algún nodo de nivel 1 (zona amarilla) y así sucesivamente. Es importante destacar que la posición geográfica de cada uno de los nodos dependerá del alcance del módulo de comunicación solamente en el

caso en que el alcance de todos los teledetectores sea mayor que el de los módulos de comunicación, en caso contrario la posición geográfica de los nodos de la red será definida en función del teledetector con menor alcance.

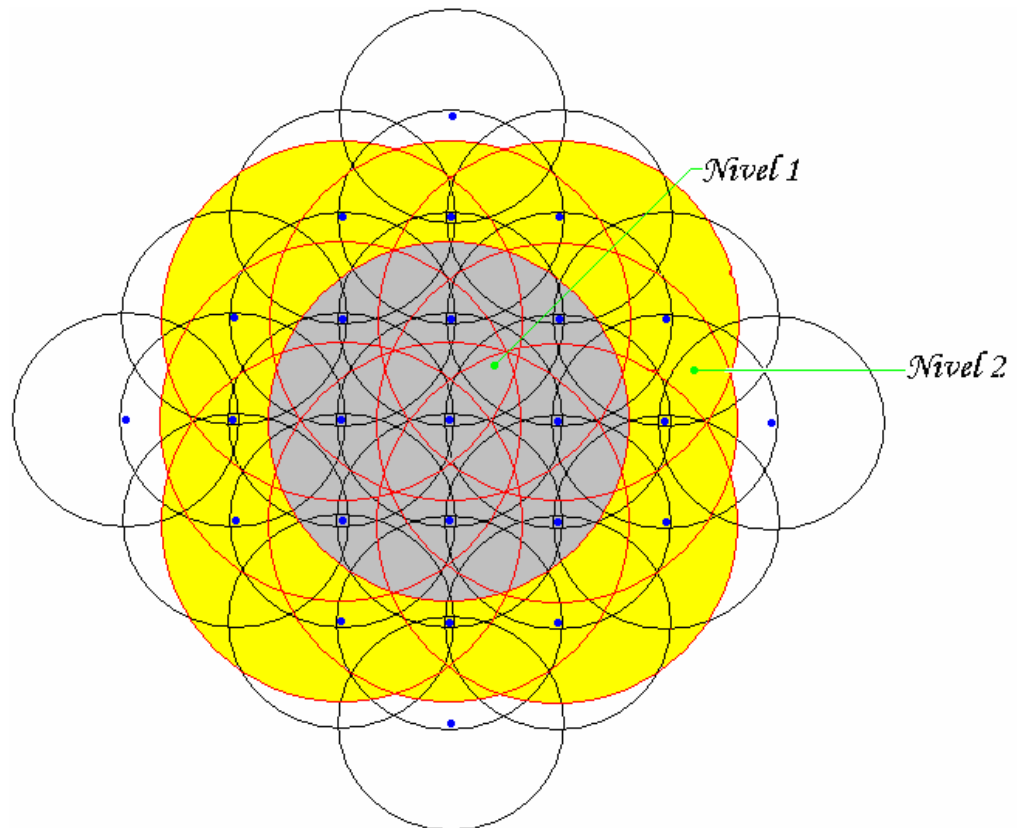


Figura 5.3: Ilustración del caso en que el alcance del módulo de comunicación es Mayor que el de los teledetectores

La principal consecuencia de esto es que puede haber “N” nodos de nivel “k” dependiendo de la ubicación que se le asigne a cada uno de los nodos, y pueden haber “N” nodos en un dominio de colisión dado, Por ejemplo en la figura 5.3 los nodos se colocan de acuerdo con el radio de alcance del teledetector que abarca un área menor (este radio se representa por medio de los círculos negros) mientras que los niveles de la red siguen siendo definidos en función del alcance del

módulo de comunicación (los cuales se representan por medio de los círculos rojos).

5.2.2 Diseño del Protocolo:

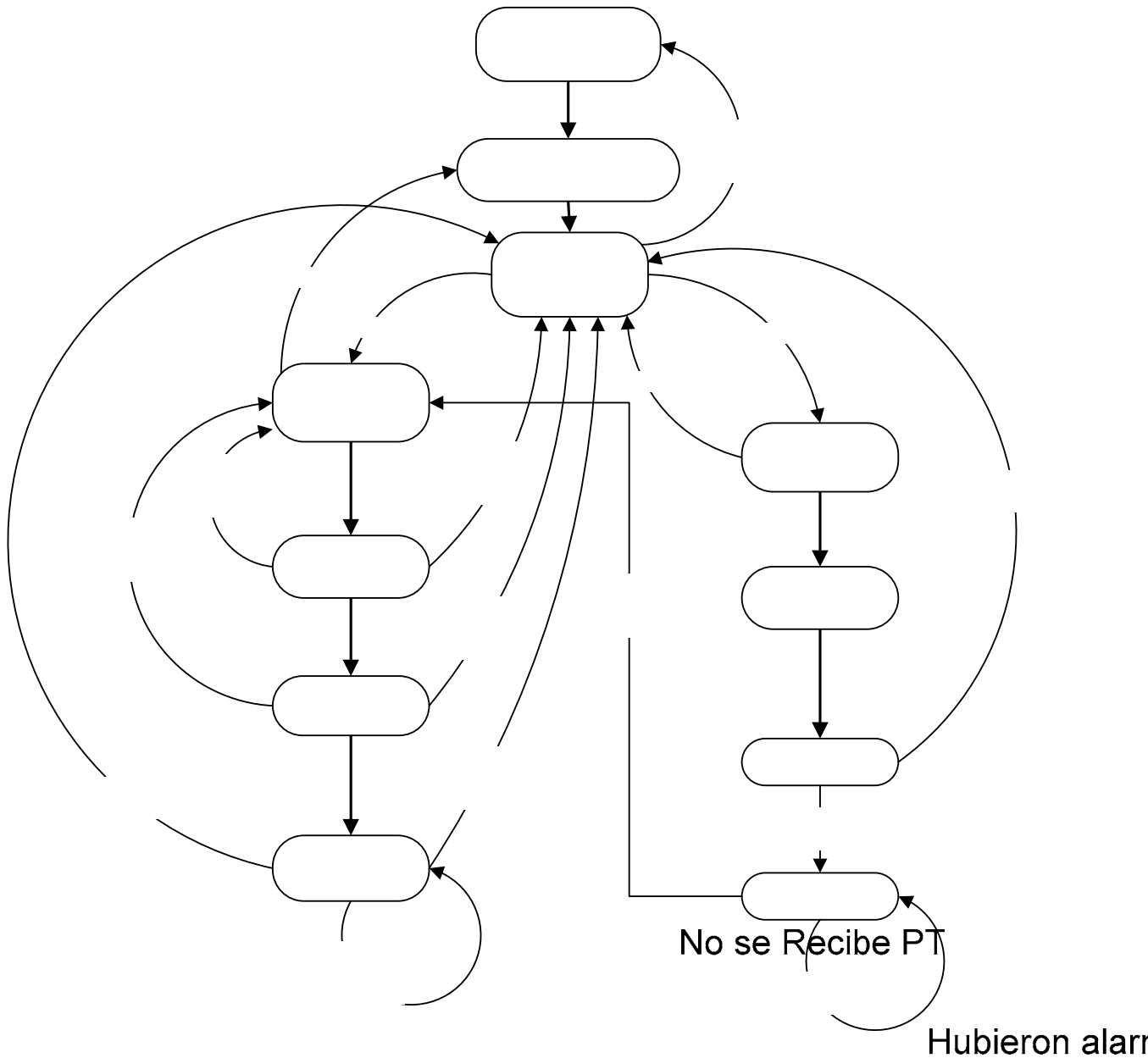


Figura 5.4: Ilustración de los estados seguidos por cada uno de los nodos del protocolo

Primer estado de espera

Escucha CTS para otro nodo y contador menor que 3

Escucha PT

Escucha para y con

Se propone que cada uno de los nodos pase a través de una serie de estados que contemplan un estado de hibernación para reducir el consumo de energía en el módulo de comunicación; Además se procura que el protocolo tenga la capacidad de configurar de manera automática la arquitectura de la red evitando de esta manera la intervención externa en el proceso de instalación y/o ampliación del sistema.

El protocolo propuesto se describe de forma gráfica en la figura 5.4, en la que se esboza la manera en la que cada uno de los nodos pasa de un estado a otro. A continuación se explica la función de cada uno de los estados del protocolo:

Inicialización de los componentes del nodo:

En este estado se inicializan la etapa de comunicación del nodo con el objetivo de asegurar un funcionamiento correcto de la misma cada vez que entra en escena.

Descubrimiento de nivel de pertenencia:

En esta etapa el nodo descubre su nivel de pertenencia en la red; esto lo hace basándose en la definición de nivel que se proporcionó anteriormente: “Se definen como nodos de nivel 1 todos los nodos que se encuentren en el alcance de la estación base, los nodos de nivel 2 serán los que se encuentren en el alcance de algún nodo de nivel 1 y así sucesivamente”. De esta definición se puede ver claramente que la pertenencia a un nivel u otro se encuentra en función del radio de alcance del transmisor del nodo de nivel inmediatamente inferior.

En este estado se escucha el medio durante un tiempo suficiente para detectar las tramas “PT¹⁰” de todos los nodos cercanos y comparar el nivel de los mismos. Una vez que se tienen los niveles de todos los nodos cercanos se selecciona el menor nivel escuchado y se sabe que el nodo que escucha se encontrará “en el nivel menor escuchado + 1”. Al finalizar este proceso se procede a enviar una trama de alarma especial que indica que el nodo entró en operación en este caso se plantea el uso de la alarma 0¹¹.

Hibernación:

En este estado no se reciben ni se transmiten datos; la etapa de radiofrecuencia se apaga completamente para conservar la energía. Este estado finaliza cuando termina el tiempo destinado al mismo¹². Una vez que finaliza se revisa si existen alarmas por enviar. Si hay alarmas se pasa al primer estado de espera y en caso contrario a la primera fase de petición de trama. Además tiene un contador que lleva el control de cuántas veces se ha pasado por este estado; Si este contador detecta que el protocolo ha pasado más de 652 veces por este estado (aproximadamente cada 2 horas con la red en reposo partiendo de un tiempo $B = 1.7$ s) y no se tienen alarmas en cola por retransmitir entonces el nodo pasa al estado de inicialización de los componentes del nodo para verificar la arquitectura de la red.

¹⁰ Este tipo de tramas se definen en el estado de “Primera fase de petición de trama

¹¹ Ver el apartado de diseño de las tramas e información de los nodos en el apartado de diseño de las tramas de alarma

¹² Este tiempo al igual que todos los tiempos a los que se hace mención en este apartado se describen con mayor detalle en la sección de diseño de los tiempos del protocolo.

Primera fase de petición de trama:

En este estado el nodo envía una trama “broadcast” preguntado a los nodos de nivel superior si tienen una trama para retransmitir (de aquí en adelante se le llamará a esta trama PT “pregunta de transmisión”) en caso de no recibir respuesta el nodo vuelve al estado de hibernación.

Segunda fase de petición de trama:

En casos de que alguna de las PT descritas en el estado anterior sea contestada (con una trama RTS), el nodo pasa a la segunda fase de petición de trama; en este estado el nodo envía una trama broadcast “CTS” para informar al primer nodo que envió el RTS que tiene el canal asignado para la transmisión de la trama, los nodos que escuchen el “CTS” se mantienen en silencio mientras se transmite el mensaje con el objetivo de evitar que las tramas colisionen.

Escucha:

Una vez que el nodo envía el CTS, espera para recibir el mensaje del nodo que solicitó el canal; si el tiempo de espera se acaba, el nodo asume que el transmisor tuvo un fallo y retorna al estado de hibernación. Si se recibe la trama a retransmitir, el nodo pasa al estado de verificación.

Verificación:

En este estado se envía una trama de “ACK” y se espera un tiempo para ver que la trama se haya recibido correctamente; si el nodo emisor de la trama no corrige el mensaje enviado se asume que el mismo se recibió correctamente y se pasa al primer estado de espera, en caso de que la trama sea corregida se guarda la nueva información de la alarma y se inicia nuevamente el estado.

Primer estado de espera:

En este estado el nodo espera por una trama PT de algún nodo de nivel inferior; en el momento en que dicha trama es escuchada el nodo pasa al segundo estado de espera. Si el nodo escucha una trama PT de un nodo de nivel inferior al inmediatamente inferior al propio, cambia su identificador de nivel para ajustarse a la nueva configuración de la red en caso de no recibir una trama de PT válida después de un tiempo definido el nodo asume que hubo un cambio en la red y vuelve al estado de descubrimiento de nivel sin borrar la memoria de alarma.

Este estado tiene una doble función ya que además de tener un papel importante en el proceso de transmisión de los datos también tiene el propósito de autoconfigurar la red en caso de que ocurra un cambio en la topología que la rodea.

Segundo estado de espera:

Después de ingresar al segundo estado de espera el nodo espera un tiempo aleatorio corto¹³ (esto con el objetivo de evitar colisiones con algún otro nodo que desee contestar la transmisión PT) y envía una trama de “RTS” para luego pasar al tercer estado de espera.

En caso de escuchar un CTS proveniente del nodo del que escuchó la trama de PT en el periodo de espera, se incrementa un contador y el nodo regresa al primer estado de espera.

Si el contador indica que el nodo ha pasado por este estado 3 veces entonces en lugar de pasar al primer estado de espera el nodo asume que hay muchos nodos encendidos y pasa al estado de hibernación sin borrar la alarma.

¹³ Este es un tiempo aleatorio discreto, y se explica con más detalle en el apartado de diseño de los tiempos del protocolo

Tercer estado de espera:

En este estado el nodo espera por un “CTS” dirigido hacia el mismo; y en caso de ser recibido el nodo pasa al cuarto estado de espera.

En caso de que no reciba la trama “CTS” el nodo incrementa un contador que indica la cantidad de veces que el nodo ha pasado por este estado y pasa al primer estado de espera. Si el contador indica que el nodo ha pasado por este estado 3 veces entonces en lugar de pasar al primer estado de espera el nodo asume que hay muchos nodos encendidos y pasa al de hibernación sin borrar la alarma.

Cuarto estado de espera:

En este estado el nodo envía la trama de alarma y espera por una trama de “ACK” (de reconocimiento de envío) que indica que la información llegó correctamente al nodo receptor; en caso de no recibirse después de un tiempo de espera corto, el nodo pasa al estado de hibernación sin limpiar la memoria de la alarma. En caso de que el nodo reciba la trama de “ACK” se verifica que la información recibida sea correcta, si no es así se retransmite la trama y se espera nuevamente el ACK (o sea se vuelve a iniciar el cuarto estado de espera), en caso de que la información sea recibida correctamente se limpia la memoria de la alarma y se pasa al estado de hibernación.

Alarma:

La ocurrencia de alarmas se estará monitoreando al final de cada estado de hibernación y se almacenarán en una tabla en el orden en que aparezcan para ser enviadas a través de la red.

En el caso de la estación base. Solamente escucha para verificar las transmisiones de los nodos de nivel uno y no detecta eventos; el diagrama de estados que se sigue en este nodo se muestra a continuación:

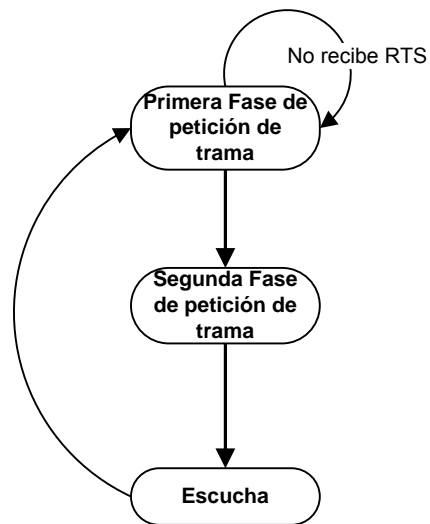


Figura 5.5: Ilustración de los estados seguidos por el nodo base

5.2.3 Diseño de las tramas e información de los nodos:

Para el diseño de las tramas se procura que las mismas tengan una cantidad mínima de bits con el propósito de minimizar el tiempo que duran las transmisiones y con esto la probabilidad de colisión así como el consumo de energía en el proceso de envío.

Además es importante tomar en cuenta que los encabezados de las tramas que son las que le dan identidad a cada tipo de trama sean diferentes de cualquier otro dato posible de información para evitar que algún nodo confunda entre el inicio de una trama y un dato común.

Las tramas requeridas para la implementación del protocolo anteriormente descrito son las siguientes:

Trama de PT:

Es importante que esta trama contenga el nivel del nodo que la emite con el objetivo de los nodos que la escuchen puedan verificar que la misma corresponde a un nodo más cercano a la base que ellos mismos además es importante que la trama contenga la dirección MAC del emisor para que el receptor sepa a quien debe hacer el RTS.

0xF1 Nivel del emisor Dirección MAC del emisor 0xF1

Trama de RTS:

La trama de RTS debe contener la dirección MAC del emisor para que el nodo que la escuche la incluya en la trama del CTS y este sepa que su solicitud fue aceptada, además debe contener el nivel del nodo emisor para facilitar el proceso de búsqueda de identidad de los nodos vecinos

0xF2 Nivel del emisor Dirección MAC del emisor 0xF2

Trama de CTS:

La trama CTS es una trama broadcast y por esto debe contener la dirección MAC del nodo a la que va dirigida para que los demás nodos sepan a quien se dirige y guarden silencio en caso de que no sea para ellos

0xF3 Nivel del emisor Dirección MAC del destinatario 0xF3

Trama de alarma:

Esta trama debe dejar claro el lugar de procedencia y el tipo de la misma es por esto que debe contener el código de la alarma así como la dirección MAC del nodo en el que se produjo.

0xF4 Código del tipo de alarma Nivel del emisor Dirección MAC de la procedencia

El campo que se asigna al código de la alarma es de un Byte dando de esta forma la capacidad de comunicar 256 diferentes tipos de información que pueden ser especificadas de acuerdo a las necesidades que se presenten en la red. Se definen cinco tipos de alarmas:

Código del tipo de alarma	Alarma asignada
0x04	Salida de operaciones del nodo
0x03	Fuego
0x02	Motosierras
0x01	Disparos
0x00	Inicio de operaciones del nodo

Tabla 5.1: Descripción de los tipos de alarmas asignadas hasta el momento

La salida de operación del nodo se define como una alarma generada por el módulo de alimentación e indica que el nodo se encuentra en un

límite de carga que hace necesario apagar el nodo para que se cargue hasta un nivel que permita su operación normal

Trama de ACK:

Esta trama debe dar la capacidad al nodo de saber si el dato fue recibido correctamente.

0xF5 Código del tipo de alarma Nivel del emisor Dirección MAC de la procedencia

5.2.3 Diseño de los tiempos del protocolo:

Con el objetivo de ahorrar energía en los nodos, se busca que el ciclo en que cada nodo mantiene encendida la etapa de radiofrecuencia sea el menor posible; es por esto que los tiempos de espera deben ser calculados de forma que no se desperdicie la energía y se asegure la pronta transmisión de las alarmas.

La base de tiempo B:

Este es el tiempo que se utilizará como base para el cálculo del resto de los tiempos y es igual al tiempo necesario para que la trama más grande sea transmitida a la máxima distancia de alcance de la red, que sea atendida y retransmitida hacia la primera estación nuevamente.

Es importante mencionar que este tiempo es directamente dependiente de la capa física que se utilice en la implementación del protocolo, por lo que será necesario medirlo cada vez que se cambie la tecnología utilizada en esta capa.

Este tiempo se calculó por medio de pruebas realizadas en la capa física actualmente utilizada en aproximadamente 350 ms. Este tiempo será utilizado en el resto del protocolo cuando hay que esperar la respuesta de uno de los nodos.

Tiempo de hibernación:

Este es uno de los tiempos más importantes del protocolo debido a que influye de manera directa en la cantidad de energía consumida en la etapa de comunicación de los nodos; entre más grande sea el tiempo destinado en este estado mayor será el ahorro en la energía del protocolo, no obstante el atraso en el proceso de envío de las tramas aumentará.

En este sentido si la red se encuentra en reposo (sin recibir señales de alarma) y se gasta un tiempo máximo aproximado de "P" (2B) en la primera fase de petición de trama, el porcentaje de ahorro de energía en el nodo con respecto de un protocolo en el que la etapa de radio (transmisión, recepción) debe mantenerse permanentemente encendida, se comportará aproximadamente como se muestra en la Figura 5.6 conforme aumenta el tiempo de hibernación¹⁴

¹⁴ Tomando como referencia que para los módulos que se están utilizando el consumo promedio de corriente aproximado en hibernación de 16 mA, en estado de escucha de 46mA, y en estado de transmisión configurado en baja potencia de 350 mA. Este resultado se explica con más detalle en el capítulo 6

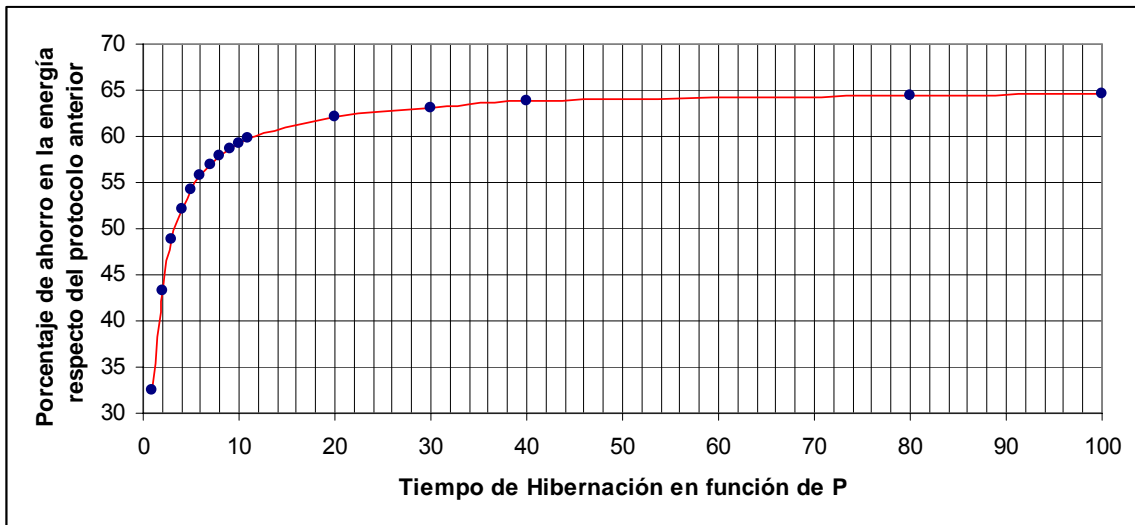


Figura 5.6: Porcentaje de ahorro en la energía para la red en reposo en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida

Se puede notar que el porcentaje de ahorro en la energía no cambia de manera drástica después de que el tiempo de hibernación alcanza un valor aproximado a los 10P como se muestra en la tabla siguiente:

T (P)	T (s)	P (s)	Potencia Promedio (W)	Potencia protocolo anterior (W)	% Ahorro
1	0.7	0.7	0.10373314	31.6645963	31.6645963
2	1.4	0.7	0.08675543	42.8488613	42.8488613
3	2.1	0.7	0.07826657	48.4409938	48.4409938
4	2.8	0.7	0.07317326	51.7962733	51.7962733
5	3.5	0.7	0.06977771	54.0331263	54.0331263
6	4.2	0.7	0.06735233	55.6308784	55.6308784
7	4.9	0.7	0.06553329	56.8291925	56.8291925
8	5.6	0.7	0.06411848	57.7612146	57.7612146
9	6.3	0.7	0.06298663	58.5068323	58.5068323
10	7	0.7	0.06206057	59.1168831	59.1168831
11	7.7	0.7	0.06128886	59.6252588	59.6252588
20	14	0.7	0.05765078	62.021887	62.021887
30	21	0.7	0.05608601	63.0526949	63.0526949
40	35	0.7	0.05479738	63.9015954	63.9015954
80	56	0.7	0.05405761	64.3889272	64.3889272
100	70	0.7	0.05380858	64.5529795	64.5529795

Tabla 5.2: Porcentaje de ahorro en la energía para la red en reposo, en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida

Es importante también destacar que para una red con N niveles en los que se programa un tiempo de hibernación “T” y la suma de todos los demás tiempos de espera es “K”, sin retransmisiones, tomando en cuenta que cada uno de los nodos opera sin sincronización; el tiempo máximo (en el peor de los casos) entre la detección de una alarma y su recepción en la estación base será:

$$T_{\max} = (N - 2) \cdot (T + K) + T_{Base} \quad (5.1)$$

Dado que el tiempo más grande contemplado en la red es el de hibernación se tiene que el tiempo entre la emisión de una alarma y su recepción en la estación base será claramente afectado por el mismo.

Basados en el criterio anterior se selecciona un tiempo de hibernación de “4P” o en nuestro caso de 2.8 segundos de forma que se asegure un ahorro de energía para la red en reposo de aproximadamente un 52% respecto del protocolo anteriormente desarrollado¹⁵ y se obtenga un retraso máximo sin tomar en cuenta colisiones de aproximadamente 4.5 minutos en una alarma generada en el nivel 53 de una red¹⁶.

Descubrimiento de nivel de pertenencia:

El tiempo que se escucha en este estado debe ser lo suficientemente largo como para que el nodo escuche las tramas PT de todos sus vecinos y al mismo tiempo no debe ser muy largo para que no se desperdicie energía.

¹⁵ Refiérase al proyecto de graduación “Diseño e implementación de una red de comunicación inalámbrica para protección ambiental en el bosque” presentado en julio del año 2005 en el Instituto tecnológico de Costa Rica.

¹⁶ Refiérase a la sección 5.3

Si se supone que la red se encuentra en reposo, entonces cada uno de los nodos transmitirá una trama PT aproximadamente cada vez que se termina el tiempo de hibernación; para asegurar que los nodos tengan el tiempo suficiente para escuchar las tramas PT de todos sus vecinos se asigna en este estado un tiempo de escucha de aproximadamente el doble del tiempo de hibernación o sea 16B, para nuestro caso aproximadamente 5.6 s.

Primer estado de espera:

En este estado, al igual que en el estado de descubrimiento de trama es necesario esperar tiempo suficiente para escuchar las PT de sus vecinos o sea 16B o para nuestro caso 5.6s.

Segundo estado de espera:

En este estado se espera un tiempo aleatorio con el propósito de evitar colisiones después de haber escuchado una trama de PT de un nodo de nivel inferior. Para el cálculo de este tiempo es importante que el tiempo aleatorio que se espera antes de enviar la trama debe ser lo más corto posible para que no se gaste mucha energía, no obstante entre más sean los posibles tiempos de transmisión en este estado, menor será la probabilidad de colisión¹⁷.

Es importante además mencionar que este estado influye directamente en el tiempo que debe ser asignado en la espera entre cada una de las tramas de "PT" de la primera fase de petición de trama y con esto influye directamente en el tiempo que debe ser asignado al estado de hibernación y al tiempo de retraso total de las tramas transmitidas.

Tomando en cuenta todo lo anterior este tiempo se define entre 0 y 5B,

¹⁷ Ver el apartado de análisis teórico del protocolo.

de esta forma el nodo tiene 6 tiempos posibles en los cuales puede intentar transmitir su trama; en $t=0$, $t=B$, $t=2B$, $t=3B$, $t=4B$ y $t=2B$ esto con el objetivo de que si otro nodo transmite en un tiempo menor el nodo que se encuentra esperando tenga tiempo suficiente para escuchar la trama de CTS y evite la colisión.

Primera fase de petición de trama:

En este estado es necesario implementar un tiempo de espera para la trama "PT", este tiempo debe ser aproximadamente $2.5 B^{18}$ para dar tiempo a que la estación transmita en el peor de los casos del tiempo de espera anteriormente descrito.

Segunda fase de petición de trama:

En este estado es importante mencionar que el tiempo que los nodos (a los que no se les asignó el canal) esperarán en caso de escuchar la trama de CTS es aproximadamente el tiempo que se calcula que dura el ciclo de transmisión de trama después de que es recibido el CTS o sea $2B$.

Estado de escucha:

En este estado se espera un tiempo de B después de haber enviado la trama de CTS para recibir el mensaje del nodo al que se le notificó, o para volver al estado de hibernación.

¹⁸ Es importante mencionar que para este caso, al igual que para todos los demás casos que se analizarán a continuación, se define un tiempo máximo de espera; No obstante si la trama esperada llega antes de que el mismo finalice el nodo seguirá adelante con el resto del proceso.

Verificación:

En este estado se espera un tiempo de B antes de dar por aceptado como correcto el mensaje recibido después de enviar la trama de ACK

Tercer estado de espera:

Al igual que en el caso anterior el nodo esperará B para recibir un CTS.

Cuarto estado de espera:

En este estado el nodo espera un tiempo B para recibir la trama de ACK.

5.2.3 Análisis Teórico del protocolo propuesto:**Retrazo de las tramas suponiendo que no hay colisión:**

Anteriormente se dijo que el retraso máximo que sufriría una trama estaba dado por la fórmula:

$$T_{\max} = (N - 2) \cdot (T + K) + T_{Base} \quad (5.2)$$

Tomando en cuenta la información teórica suministrada en el apartado de diseño de los tiempos de la red podemos decir que el tiempo de retraso para que una trama se transmita de un nivel al siguiente estará dado por:

- El tiempo transcurrido desde que se da la alarma y hasta que el nodo se despierte para atenderla ($0B_{\min}, 4B_{\max}$)
- + tiempo transcurrido entre que el nodo entra al primer estado de espera y que escuche una trama de PT ($0B_{\min}, 4B_{\max}$)
- + el tiempo aleatorio seleccionado en el segundo estado de espera para enviar la trama de RTS ($0B_{\min}, 5B_{\max}$)

- + el tiempo que tarda en escuchar la trama CTS (aprox = B)
- + el tiempo que tarde en escuchar la trama de ACK (aprox =B):

$$\begin{aligned} T_{\max imo} &\approx 15B \\ T_{\min imo} &\approx 2B \end{aligned} \quad (5.3)$$

Esto excepto en la transición entre el nivel 1 y el nivel 0 (o sea la base) en la que el retraso máximo será de 2B y el mínimo de 0B, esto quiere decir que el tiempo de espera en caso de no haber colisiones varía entre un rango de:

$$\begin{aligned} T_{\max imo} &\approx (N - 2) \cdot 15B + 2B \\ T_{\min imo} &\approx (N - 2) \cdot 2B \end{aligned} \quad (5.4)$$

Para una red en la que se detecta una alarma en el nivel N.

Esto quiere decir que para una red en la que se detecta una alarma en el nivel 50, con un tiempo B de 0.35 s el retraso máximo que sufrirá la trama en llegar a la base será de aproximadamente 4.21 minutos, y el mínimo de aproximadamente 33.6 segundos.

Este retraso es aceptable tomado en cuenta que la atención de la alarma después de recibida podría tomar horas.

Además de las formulas anteriores se puede ver que este tiempo es proporcional a la base de tiempo B que dependerá de la capa física de la red.

En la figura 5.7 se obtienen los valores de tiempo de retraso máximo y mínimo para alarmas que ocurren en diferentes niveles con un B de 0.35s:

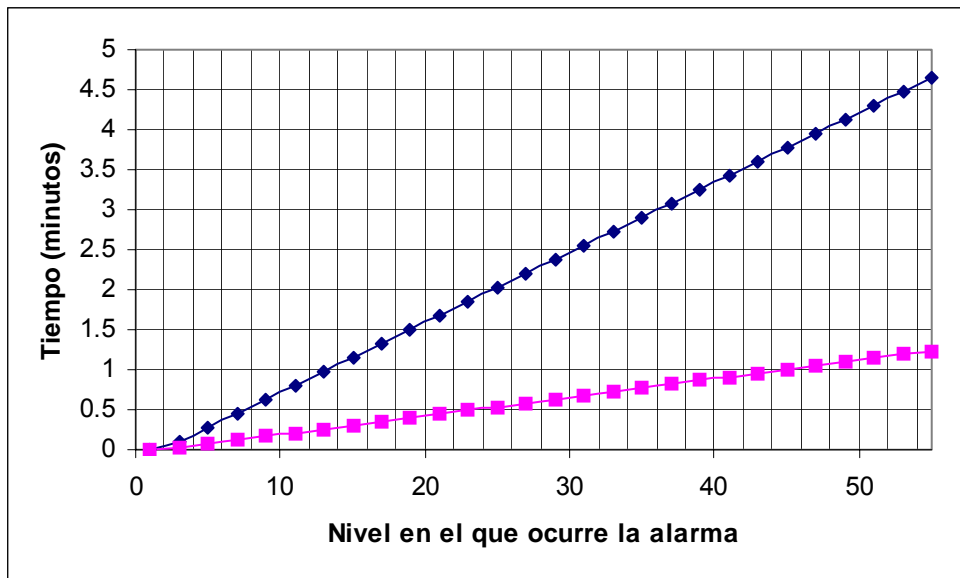


Figura 5.7: Ilustración del tiempo de retardo mínimo y máximo que tarda una alarma que se da en un nivel dado en llegar a la base

De este gráfico se puede notar que la brecha entre el valor máximo y el mínimo de retraso aumenta de forma aproximadamente lineal, como se muestra en la figura 5.8, conforme aumenta el nivel en el que se da un evento:

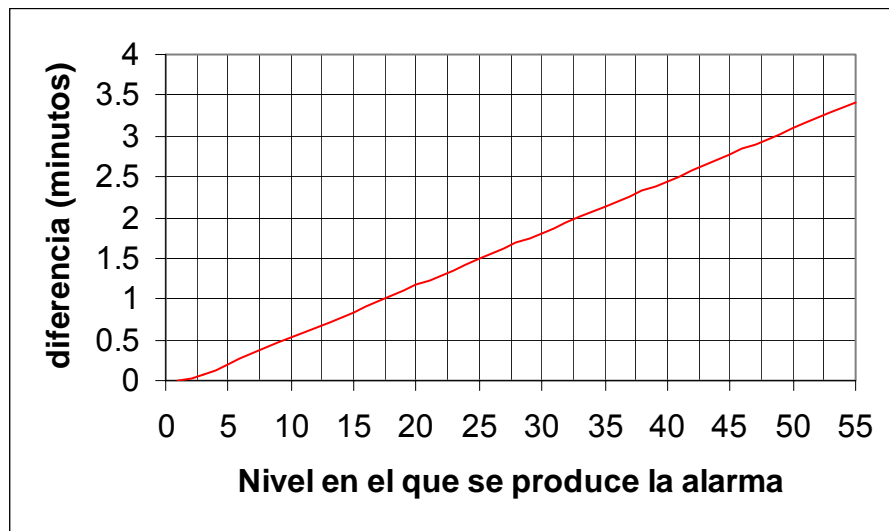


Figura 5.8: Ilustración de la diferencia entre el tiempo de retardo mínimo y máximo conforme aumenta el nivel en el que se produce una alarma

Esta característica hace que entre más alto sea el nivel de la red en el que se produce un evento sea más difícil estimar el tiempo exacto en el que se produjo el mismo.

Nivel	Retrazo máximo (s)	Retrazo mínimo (s)	Tiempo estimado que ocurrió la alarma (s)	desde Precisión (± s)
1	0.7	0	0.35	0.35
3	5.95	1.4	3.675	2.275
5	16.45	4.2	10.325	6.125
10	42.7	11.2	26.95	15.75
20	95.2	25.2	60.2	35
30	147.7	39.2	93.45	54.25
50	252.7	67.2	159.95	92.75
60	305.2	81.2	193.2	112

Tabla 5.3: Aumento de la incertidumbre en la estimación del tiempo en el que ocurrió un evento conforme aumenta el nivel en el que se produce para un B de 1.7 s

Análisis probabilístico del retraso de las tramas con la red en reposo:

Del apartado anterior se puede ver que el tiempo de retraso para que una trama se transmitida de un nivel al siguiente estará dado por:

$$T = A + B + C + D + E \quad (5.5)$$

A = El tiempo transcurrido desde que se da la alarma y hasta que el nodo se despierte para atenderla ($0B_{\text{mín}}, 4B_{\text{máx}}$)

B = El tiempo transcurrido entre que el nodo entra al primer estado de espera y que escuche una trama de PT ($0B_{\text{mín}}, 4B_{\text{máx}}$)

C = El tiempo aleatorio seleccionado en el segundo estado de espera para enviar la trama de RTS ($0B_{\text{mín}}, 5B_{\text{máx}}$)

D = El tiempo que tarda en escuchar la trama CTS (aprox = B)

E = El tiempo que tarde el Nodo en escuchar la trama de ACK (aprox=B):

Esto exceptuando la transición entre el nivel 0 y el nivel 1.

Análisis de los tiempos A y B

Cuando la red se encuentra en reposo cada nodo envía una trama PT periódicamente cada “ $T^{19}+P^{20}$ ” segundos (exceptuando la base que lo hace cada $(T+P)/10$ s), de esta manera se tiene que para una red en reposo se puede decir que la probabilidad de que un nodo salga del estado de hibernación en un periodo de tiempo de “ $T+P$ ” segundos es igual a 1.

De la afirmación anterior se deduce que la función de probabilidad de que un nodo salga de hibernación en un intervalo de tiempo dado entre a y b será:

$$\psi = \int_a^b \frac{1}{(T + P)} \cdot dx \quad (5.6)$$

$$\psi = \frac{1}{(T + P)} \cdot (b - a) \quad (5.7)$$

De donde se puede ver que la probabilidad de que el nodo se “despierte” o salga del estado de hibernación en un periodo de tiempo dado entre a y b es constante e igual a ψ .

Análisis del tiempo C:

A diferencia de los tiempos A y B, el tiempo C es discreto y está diseñado de esta forma para evitar colisiones entre las tramas.

Si se supone que el generador de números aleatorios que se utiliza en la implementación del segundo estado de espera tiene una distribución constante, entonces la probabilidad de que el nodo seleccione

¹⁹ Tiempo que se asigna en el estado de hibernación.

²⁰ Tiempo que se gasta en el proceso de petición de trama

cualquiera de las opciones de espera ($0B$, $-5B$) es la misma e igual a 16.66%

Análisis de los tiempos D y E

Dado que estos tiempos son constantes la consecuencia que tienen en la probabilidad del retraso en la transición de dos nodos de nivel superior a 1 es que establecen un tiempo mínimo de retraso de forma que:

La probabilidad de que el tiempo de espera sea menos que $D+E = 0$

Probabilidad de retraso de las tramas en las transiciones de los nodos de nivel superior a 1

Tomando en cuenta que cada uno de los nodos trabaja de forma independiente se pueden modelar los tiempos de espera de cada una de las transiciones como eventos independientes, de esta forma la probabilidad de retraso debida a los tiempo A y B se puede ver como una función de probabilidad uniforme continua como la que se muestra en la figura 5.9:

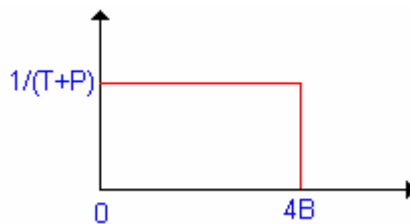


Figura 5.9: Función de probabilidad de los tiempos A y B

Se parte de que los tiempos A y B son dos variables aleatorias independientes con funciones de densidad de probabilidad iguales a la que se muestra en la figura 5.9 y a las que llamaremos a y b,

respectivamente; Entonces la densidad de probabilidad de la suma de los tiempos aleatorios $A + B$ vendrá dada por la convolución $a * b$.²¹

$$a * b = \int a(\tau) \cdot b(t - \tau) d\tau \quad (5.8)$$

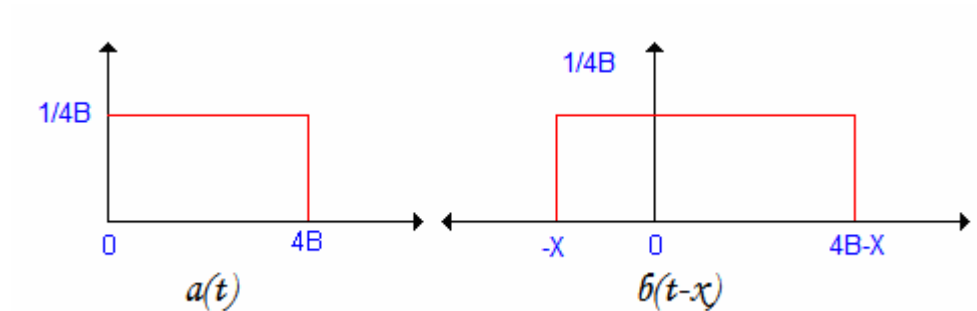


Figura 5.10: Componentes de la convolución de las funciones de densidad de probabilidad a y b

Se puede dividir la integral de manera que:

$$a * b = \left(\frac{1}{4B}\right)^2 \cdot t \rightarrow 0 \leq t \leq 4B \quad (5.9)$$

$$a * b = \left(\frac{1}{4B}\right)^2 \cdot (8B - t) \rightarrow 4B \leq t \leq 8B$$

Partiendo de lo anterior se sabe que la función de probabilidad que modela el retraso debido a los tiempos $A+B$ se puede modelar como se muestra en la Figura 5.11:

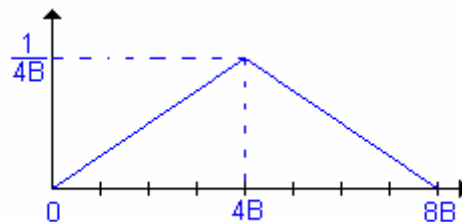


Figura 5.11: Función de probabilidad del retraso debido a $A+B$

²¹ Ver apéndice A3

Tomando en cuenta la influencia de los tiempos D y E se tiene que la función de probabilidad del retraso será :

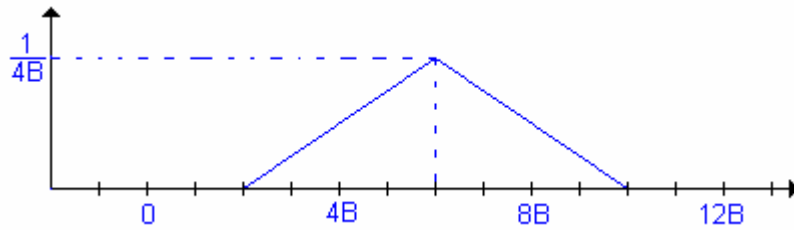


Figura 5.12: Función de probabilidad del retraso debido a A+B tomando en cuenta Los tiempos D y E

Si se toma en cuenta el retraso aleatorio de C se tiene que la función de probabilidad será igual a la suma de las funciones de probabilidad para las diferentes selecciones posibles, ver figura 5.13:

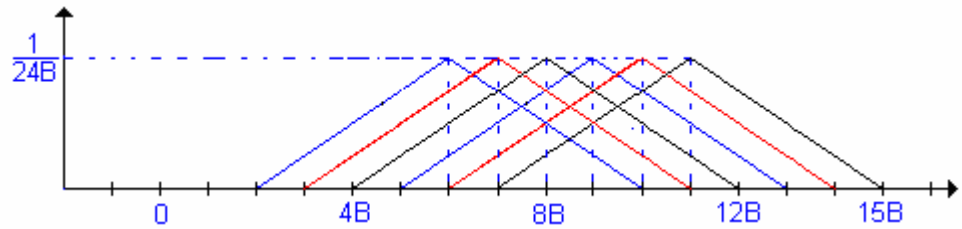


Figura 5.13: Funciones de probabilidad para el retraso para cada una de las posibilidades de selección de C

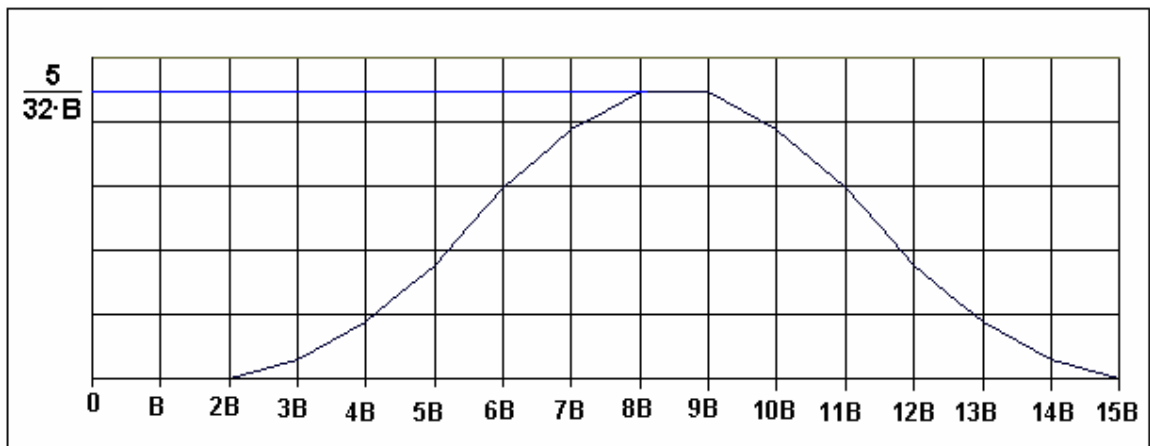


Figura 5.14: Función de probabilidad para el retraso de las tramas en las transiciones entre los nodos de nivel superior a 1 en función de B

De la función anterior se puede calcular la probabilidad de que el retraso se encuentre en un intervalo dado entre “a” y “b” por medio de:

$$\int_a^b f_x(X)dx \quad (5.10)$$

Partiendo de los resultados expuestos anteriormente se puede hacer uso de Matlab para encontrar un polinomio de interpolación y encontrar la probabilidad de que la latencia de la trama se encuentre en los diferentes intervalos.

Así las cosas se tiene que la probabilidad de que el retraso en la transmisión de las tramas se encuentre entre los diferentes intervalos posibles queda como se muestra en la tabla siguiente:

Intervalo	Probabilidad
(2B - 3B) = (14B - 15B)	0.5%
(3B - 4B) = (13B - 14B)	1.98%
(4B - 5B) = (12B - 13B)	4.6875%
(5B - 6B) = (11B - 12B)	8,33%
(6B - 7B) = (10B - 11B)	11,45%
(7B - 8B) = (9B - 10B)	14.6
(8B - 9B)	15.63%

Tabla 5.4: Porcentaje de probabilidad de que el retardo de transmisión entre 2 nodos de nivel superior a 1 se encuentre en un intervalo dado

Es importante mencionar que la tabla de probabilidades es independiente del tiempo B; además se puede ver que la mayor probabilidad recae en el intervalo entre 6B y 11B con un 67.73%.

No obstante que las probabilidades en los diferentes intervalos sean independientes de la base de tiempo B conforme la misma disminuya el atraso en las tramas lo hará de una manera proporcional.

También es importante destacar que conforme el número de posibles valores del tiempo “C” aumente, el intervalo de valores posibles en el retraso aumentará hacia la derecha e igualmente lo hará la curva de probabilidades.

Análisis de la probabilidad de colisión:

En general pueden existir colisiones en el protocolo en diferentes puntos debido a diferentes causas. En la presente sección se abordarán las que se han considerado más importantes en función de los objetivos propuestos para el presente proyecto.

Probabilidad de colisión de las tramas de PT:

La probabilidad de que ocurra una colisión en las tramas de PT estará en función de la cantidad de nodos que se encuentren en el mismo dominio de colisión, el tiempo que se asigna al estado de hibernación y la duración en el envío de la trama.

Así si se asigna un tiempo de Hibernación igual “T” y se sabe que las tramas de PT tardan un tiempo “F” para ser transmitidas, y además se asume que la probabilidad de que haya un nodo inicialmente encendido es constante para todo el intervalo de tiempo entre tramas de PT, se tiene que la probabilidad de colisión de tramas de PT es igual a:

$$Pr_{obPT} = \frac{F}{T} \cdot \#de_nodos \quad (5.9)$$

De la ecuación 5.9 podemos ver que entre mayor sea el tiempo destinado al estado de hibernación no solo aumenta el porcentaje en el ahorro de energía sino que la probabilidad de colisión en las tramas de PT disminuye.

Probabilidad de que ocurra una colisión en las tramas de RTS

La probabilidad de que ocurran dos eventos o que un evento sea escuchado por dos nodos distintos en el mismo nivel de la red y que además estos nodos cuenten con uno o más nodos vecinos de nivel inferior en común, y que además despierten en un tiempo adecuado para que escuchen la misma trama de PT de nivel inferior se asume pequeña, y para calcularla sería necesario tomar en cuenta una serie de factores de la topología de la red, ubicación física de los nodos, alcance de la etapa de radio, alcance de los teledetectores y características específicas de la región en la que se encuentra instalada la red.

Es por esto que de aquí en adelante solamente asumiremos esta probabilidad como “H” para cada uno de los casos en los que resulta necesario tomarla en cuenta.

Dado que se tengan dos nodos en espera de transmitir un evento y ambos escuchan la misma trama de PT la probabilidad de colisión vendrá dada por la cantidad de posibles tiempos de transmisión de la trama de “RTS”²². En este caso la probabilidad de que ocurra una colisión dado que los nodos tienen seis tiempos posibles en los que pueden transmitir será:

$$\frac{6}{6^2} = 0.16667$$

²² Definidos en el segundo estado de espera

en general la probabilidad de que ocurra una colisión con dos nodos tratando de transmitir será : $H * 0.16667$

En caso de que existan tres nodos tratando de transmitir un la probabilidad de que ocurra una colisión será:

Probabilidad de colisión con tres nodos $=H* 0.333+ 6/63$

En caso de existir siete o más nodos tratando de transmitir la probabilidad de colisión será de un 100%. Esto se repetirá cada vez que se tengan "x+1" nodos tratando de transmitir cuando existen "x" posibles tiempos de transmisión en el segundo estado de espera.

Esta característica hace que no sea recomendable que se pongan más de dos 4 nodos con un solo vecino en común capaz de retransmitir hacia la base los eventos detectados por los mismos ya que de lo contrario la probabilidad de que la alarma sea transmitida sin que ocurra una colisión sería menor que el 66.6%.

Una forma simple de reducir esta probabilidad es aumentado los posibles tiempos de transmisión asignados en el segundo estado de espera, no obstante esto degradaría la latencia de la red (como se mencionó en el apartado anterior).

No obstante, la probabilidad de que ocurra la situación en que dos o más nodos vecinos del mismo nivel compitan por contestar una trama "PT", se asume muy pequeña por lo que esto no resulta en un problema grave.

5.3 Descripción del hardware

Para el presente proyecto se hace uso del hardware desarrollado en el trabajo realizado anteriormente [1].

El hardware se puede describir en bloques como:

Para los nodos de la red:

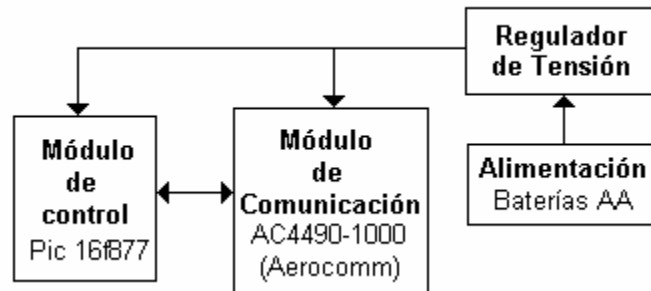


Figura 5.15: Diagrama de bloques del hardware para los nodos de la Red

Para el nodo de la Base

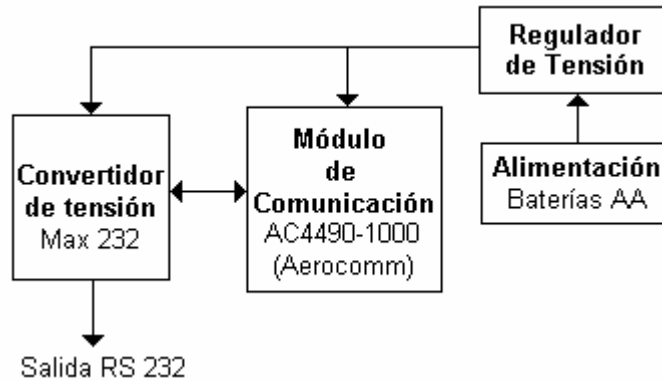


Figura 5.16: Diagrama de bloques del hardware para el nodo de base

Dado que para el presente diseño no se modificó esta etapa, no será descrita a profundidad.

Para más información refiérase a [1].

5.4 Descripción del Software

Dado que el algoritmo implementado en el presente proyecto es cubierto con detalle en la sección 5.2 del presente informe, en esta sección se enfocarán aspectos más generales relacionados con la estructura, la filosofía de implementación del software.

5.4.1 Estructura General del programa

Para la implementación del software se sigue una filosofía general que trata de cumplir con dos objetivos principales:

- Lograr hasta donde sea posible la separación de las capas del modelo de interconexión de sistemas abiertos en la programación, para facilitar la implementación de mejoras sobre el protocolo.
- Estructurar la programación mediante el uso de rutinas y macros, separando hasta donde sea posible las estructuras de control de flujo de los estados del programa para lograr de esta forma la orientación a objetos de una manera más clara y facilitando la comprensión del código.

De esta manera el software desarrollado en los nodos se divide en Cuatro partes principales:

- Rutinas generales del programa
- Estados del Protocolo
- Control del flujo del protocolo
- Interrupción del puerto serial RS 232

Rutinas Generales del programa:

En esta sección se colocan las rutinas que son utilizadas por los estados del protocolo, es en esta etapa en la que se hace una división conceptual importante en lo que se refiere a la encapsulación de las tramas ya que existe una rutina para enviar cada una de las diferentes tramas. A pesar de que estas rutinas pueden parecer triviales y hasta cierto punto restan eficiencia al programa, separan las funciones de encapsulación del resto del código lo cual hace que la implementación sea más acorde con la división de capas del modelo de interconexión de sistemas abiertos.

Estados del protocolo

En esta sección se programan cada uno de los estados expuestos en el capítulo 5 del presente informe de manera que cada uno es llamado en la sección de control de flujo como un método facilitando la comprensión del protocolo. Cada una de las acciones que se realizan en los estados se explican con detalle en la sección 5.2.2 del

Control de flujo del protocolo

Una vez que se tienen programados todos los estados del protocolo, en esta sección se utilizan variables booleanas para decidir el siguiente estado al que salta el programa de la manera que se muestra en la figura 5.4

Interrupción del puerto serial RS 232

El manejo de los datos entrantes de la red se hace por medio de una interrupción que controla el puerto serial, de la manera que se muestra en el diagrama siguiente:

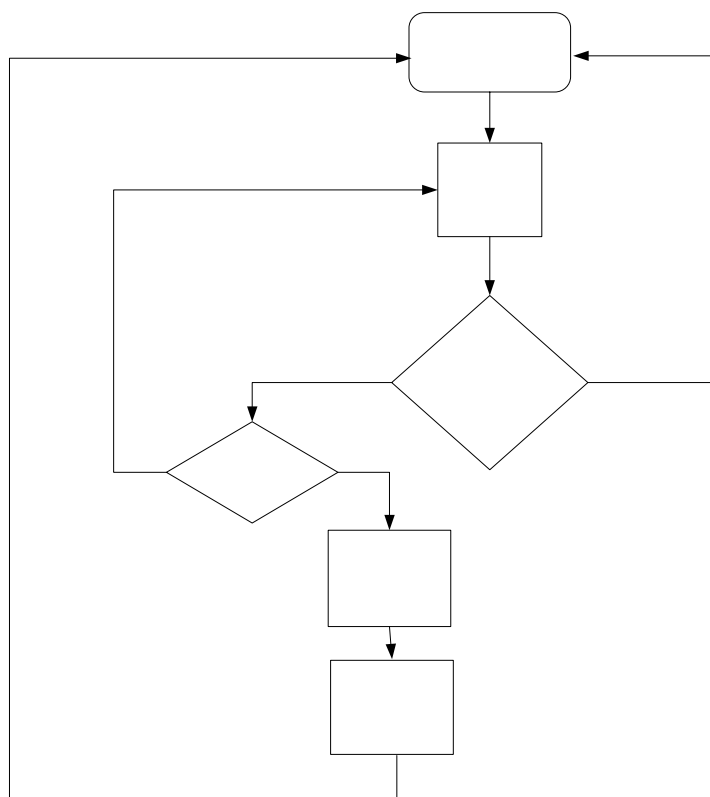


Figura 5.17: Diagrama de flujo de la interrupción del puerto serie

De esta forma se simula un proceso paralelo de decodificación de los datos entrantes. Dicho de otra manera los datos se clasifican y se envían a las capas superiores para ser procesados.

5.4.2: Funciones del modelo OSI implementadas

No Se espe trama rec

Capa de Red:

Las funciones que se asocian con la capa de red del modelo OSI son aquellas que se encargan del enrutamiento de los datos y el mantenimiento de las tablas de enrutamiento de los datos.

Con respecto al enrutamiento este se hace en función del nivel de los nodos el cual se configura en los estados de descubrimiento de nivel y el primer estado de espera.

La asignación del nivel de cada uno de los nodos se hace en función de la cercanía que los mismos tienen a la base, o un otras palabras la métrica utilizada por el protocolo es la cantidad de saltos hasta llegar a la base. Esta es una buena métrica si se parte del supuesto de que todos los nodos tienen el mismo ancho de banda.

La manera en la que esta capa es implementada no requiere el mantenimiento de tablas de enrutamiento que consumen memoria y requieren de un mayor consumo de energía y gasto de ancho de banda en el proceso de refrescamiento.

Capa de enlace de datos

La capa de enlace de datos proporciona un tránsito de datos fiable a través de un enlace físico. De este modo, la capa de enlace de datos se ocupa del direccionamiento físico de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control de flujo.

En el protocolo implementado la administración del canal se hace por medio de la implementación de tramas de ACK, y RTS además de tiempos de espera para evitar colisiones de forma similar a la que se utiliza en el Protocolo MACAW [2] y que se explicó en el apartado 5.1.

Además se implementa el direccionamiento físico de los nodos por medio de tres bytes asignados a cada uno, y que tienen la función de identificar al nodo del cual proviene la información transmitida.

El control de errores implementado evita el uso de algoritmos de CRC, o Hamming que consumen capacidad de procesamiento y atrasan el proceso de transmisión sino que se implementa por medio de una trama de ACK que asegura la confiabilidad de los datos recibidos.

Capítulo 6 : Análisis de resultados

6.1 Ahorro energético:

La primera de las ventajas del protocolo es el ahorro energético que representa respecto del utilizado anteriormente; esto debido a que en el caso anterior cada uno de los nodos tenía que mantener la etapa de radio transmisión permanentemente encendida para que el mismo funcionara correctamente.

El consumo promedio de corriente de los módulos de comunicación se midió colocando un amperímetro en serie con la batería del módulo de comunicación y los resultados obtenidos se describen en la tabla siguiente:

Estado	Corriente promedio
Escucha	46 mA
Transmisión	350 mA
Deep sleep	16 mA

Tabla 6.1: Corriente promedio en los diferentes estados de la etapa de comunicación

Tomando en cuenta que en el protocolo anterior mientras la red se encuentra en reposo el módulo de comunicación se encuentra en estado de escucha y que el voltaje aplicado en el mismo es de 3.3 V se tiene que el consumo de potencia es constante, y la potencia promedio será de 151,8 mW.

En el caso de la red en reposo con el protocolo implementado el consumo de potencia no será constante debido a que el módulo de comunicación pasará por diferentes estados a lo largo de cada uno de los ciclos de petición de trama e hibernación; de esta manera el consumo de energía en cada uno de

los ciclos puede describirse aproximadamente como se muestra en el siguiente gráfico:

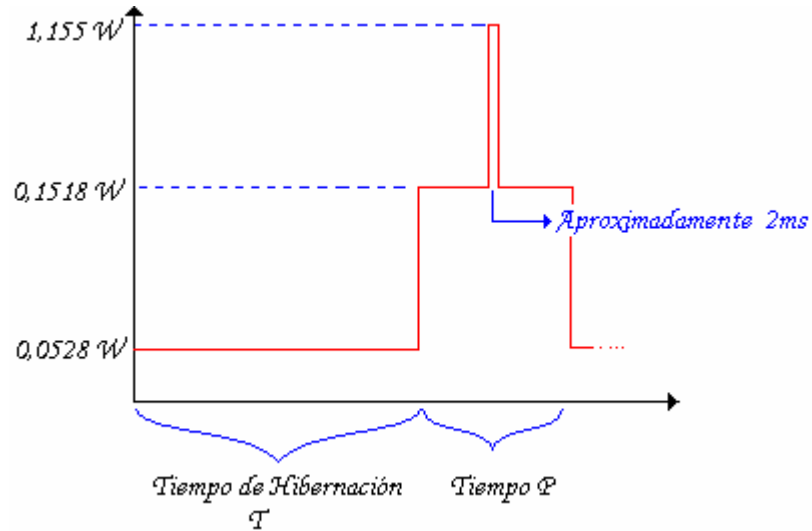


Figura 6.1: Gráfico aproximado del consumo de potencia esperado de la red en reposo utilizando el protocolo propuesto

Es importante mencionar que debido a los cambios en el consumo de potencia y de los tiempos en que se mantienen dichos cambios no fue posible medir de forma directa la potencia promedio; no obstante con base en el gráfico anterior se puede ver que la potencia promedio será:

$$P_{promedio} = \frac{0,0528 \cdot T + 0,1518 \cdot (P - 0,002) + 1,155 \cdot 0,002}{T + P} \quad (6.1)$$

De la deducción anterior se puede ver que la potencia consumida será dependiente de la proporción entre los tiempos P y T.

En la tabla siguiente se puede ver una comparación entre la potencia promedio utilizada por el protocolo anterior, y la potencia utilizada por el presente protocolo de acuerdo con la proporción entre el tiempo T y P

T / P	Potencia	Potencia	% Ahorro
	Promedio (W)	protocolo anterior (W)	
1	0,10244753	0.1518	32,511509
2	0,08589835	0.1518	43,4134697
3	0,07762376	0.1518	48,8644501
4	0,07265901	0.1518	52,1350384
5	0,06934918	0.1518	54,3154305
6	0,06698501	0.1518	55,8728535
7	0,06521188	0.1518	57,0409207
8	0,06383278	0.1518	57,9494174
9	0,06272951	0.1518	58,6762148
10	0,06182682	0.1518	59,2708672
11	0,06107459	0.1518	59,7664109
20	0,05752834	0.1518	62,1025454
30	0,05600307	0.1518	63,1073344
40	0,05474696	0.1518	63,9348077
80	0,05402586	0.1518	64,4098387
100	0,05378312	0.1518	64,5697501

Tabla 6.2: Porcentaje de ahorro en la energía para la red en reposo, en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida

Como se mencionó anteriormente el ahorro en la energía consumida crece conforme aumenta la proporción entre el tiempo T y el tiempo P.

6.2 Auto configuración:

En la versión anterior del protocolo era necesario programar cada uno de los nodos con sus vecinos inferiores y sus vecinos del mismo nivel; esta característica hace que para agregar un nodo nuevo en la red sea necesario reprogramar algunos de los nodos existentes en la misma para que el nuevo dispositivo sea aceptado. Esto requiere de un técnico capacitado que intervenga cada vez que se le quieran hacer cambios o que se quiera ampliar una red que ya se encuentre en funcionamiento, tomando en cuenta que se desea poner los dispositivos en las copas de los árboles altos de los bosques

tropicales, esta característica no solamente representa un gasto adicional para la entidad protectora del bosque sino que podría requerir de todo un despliegue de recursos para recuperar los nodos que sea necesario reconfigurar e instalarlos nuevamente en el sitio que les corresponda.

El protocolo implementado plantea una mejora significativa en este sentido ya que cuando se quiera agregar un nodo en la red no es necesario hacer cambios en la red ya existente ni tampoco se requiere de tomar en cuenta factores de topología de la red para la incorporación del nuevo componente sino que solamente es necesario encenderlo para comprobar que el mismo tenga comunicación con alguno de los nodos ya existentes en la red y de esta manera quedará en funcionamiento.

Las pruebas realizadas para comprobar el funcionamiento de este rubro se hacen utilizando la estación base, controlada con un programa de simulación realizado en matlab que simula el funcionamiento de un nodo de nivel 2; al encender un nodo en la cercanía de la base se obtiene el resultado que se muestra en la figura 6.2:

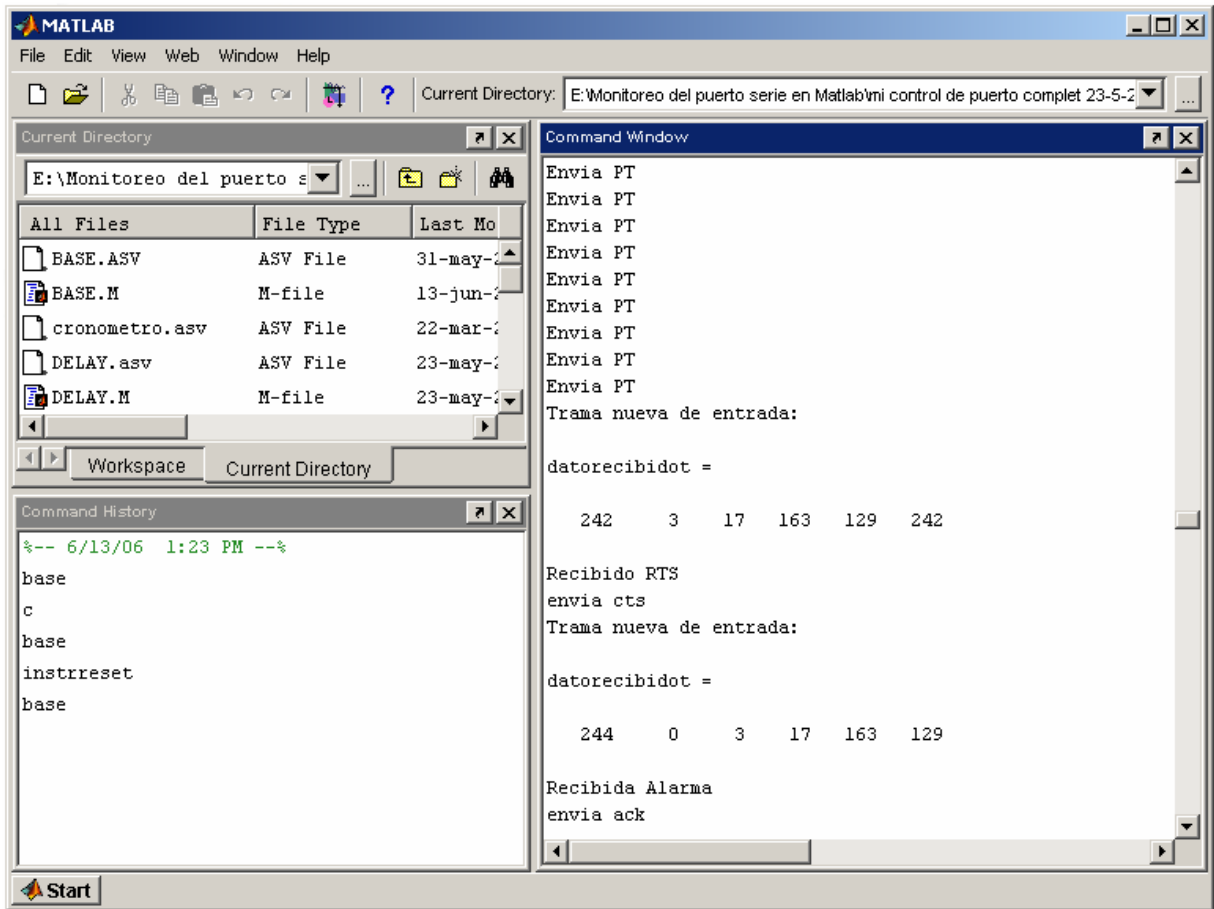


Figura 6.2: Ilustración del comportamiento obtenido en el proceso de configuración automática

En esta figura se puede ver que una vez que el tiempo de escucha de tramas PT termina, el nodo responde con una trama de petición del canal y se da la transmisión de una trama de alarma exitosamente, anunciando a la base que se configuró correctamente con un nivel 3.

Se Utiliza el mismo programa en Matlab para simular un nodo de nivel 9 que envía una alarma de tipo 0 y luego se programa la base para que en lugar de transmitir tramas de PT correspondientes a un nodo de nivel 2 envíe tramas de PT correspondientes a un nodo de nivel 3, el nodo después de un tiempo de esperar que algún nodo de nivel inferior envíe una trama de PT asume que ocurrió un problema en sus vecinos de nivel inferior y aumenta su nivel de 3 a

4; una vez en este nivel se encuentra en capacidad de escuchar las tramas que envía la base como tramas de PT válidas de acuerdo con el criterio la métrica propuesta para seleccionar la ruta (cantidad de saltos) y envía la alarma a la base que se encontraba operando como nodo de nivel 3.

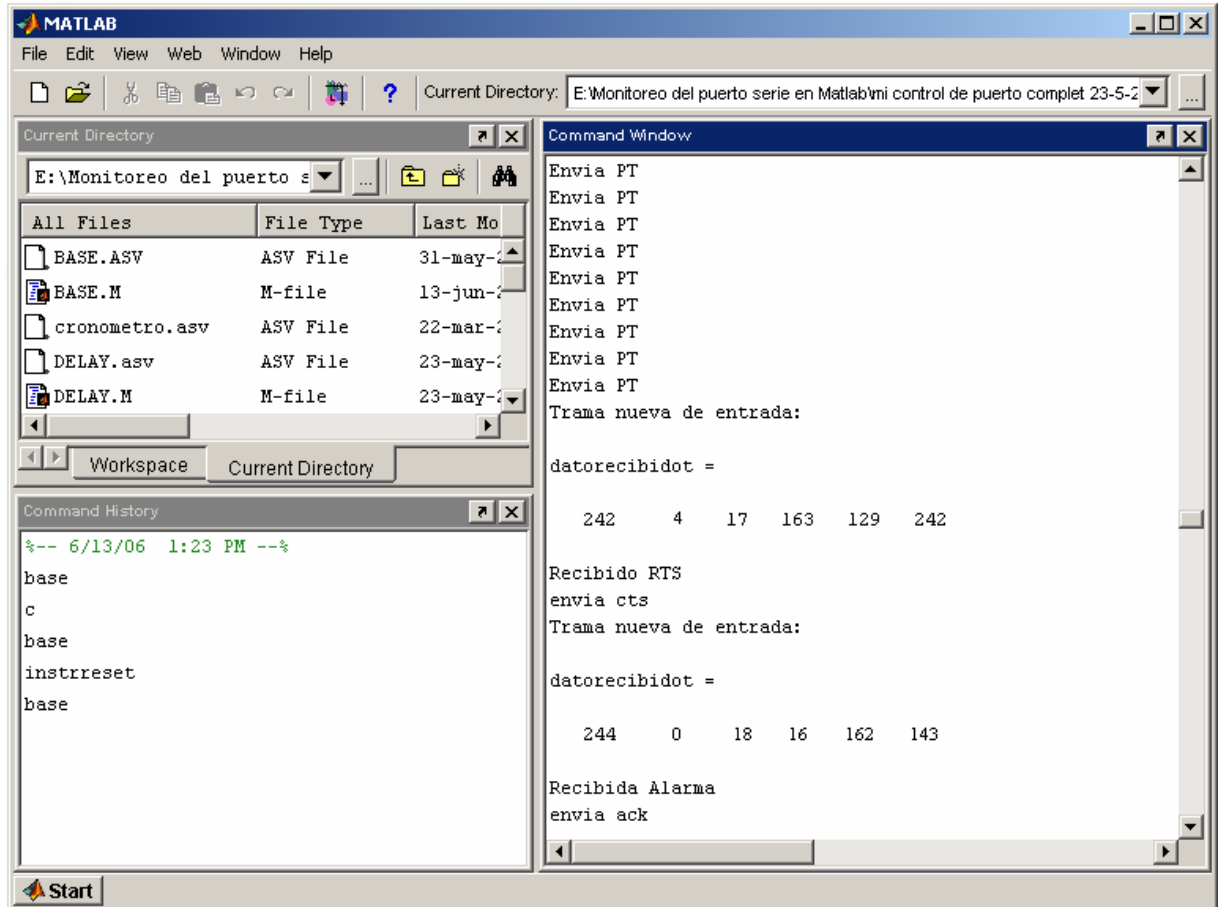


Figura 6.3: Cambio en la configuración de los nodos un cambio en la red

Además de los mecanismos anteriormente expuestos para la configuración automática de los niveles de los nodos de red, cada vez que un nodo escucha una trama PT de nivel inferior al inmediatamente inferior al propio el nodo ajusta su nivel para ajustarse a la nueva configuración

6.3 Confiabilidad:

La aparición de tramas de reconocimiento, "ACK" hace que se asegure que la información enviada por la red sea recibida en su destino sin sufrir modificaciones, no obstante presenta problemas en los casos en los que se corrompe la trama de ACK; en este caso el nodo que envía la alarma asume que la información no fue recibida correctamente debido a que no recibe al ACK y el nodo que la recibe asume que la recibió correctamente porque no recibe corrección.

En estos casos el nodo emisor regresa a estado de hibernación y retransmite la alarma haciendo que puedan existir tramas de alarma que se envíen en más de una ocasión a la base.

Para solucionar este problema se propone un control en las capas superiores de la base en la que en caso de recibir dos tramas de alarma exactamente iguales en un lapso de tiempo corto a definir, se ignore la trama duplicada.

6.4 Escalabilidad:

La propiedad de auto configuración anteriormente mencionada, facilita la adición de nuevos componentes a la red. Además el protocolo no contempla restricciones en cuanto al número de Nodos que pueden coexistir en cada uno de los niveles lo cual conlleva a la implementación de redes que pueden crecer con facilidad.

6.5 Independencia:

Una de las ventajas que presenta el presente trabajo es que no es dependiente de ninguna tecnología de capa Física en la red sino que la misma podría ser cambiada sin que esto modifique la filosofía del protocolo.

Capítulo 7 : Conclusiones y recomendaciones

7.1 Conclusiones

- El tiempo definido como B (tiempo de respuesta mínimo de la red) depende de la capa física de la Red.
- El consumo de potencia en los nodos con el protocolo propuesto es de relación inversa al tiempo asignado en el estado hibernación y de relación directa a la base de tiempo B.
- El ahorro energético obtenido con la red en reposo respecto del protocolo anteriormente desarrollado [1], es de aproximadamente un 52%.
- Los nodos se reconfiguran correctamente ante cambios en la red.
- El protocolo hace un uso mínimo de las capacidades del modulo de comunicación.
- Dependiendo de la realidad ambiental en la que trabaje una red (o sea clase de alarma más probable) y del tiempo de respuesta del guardabosques es posible programar diferentes tiempos de hibernación.

7.2 Recomendaciones

- Es recomendable para trabajos posteriores desarrollar la implementación en una FPGA con miras en integrar todas las etapas del proyecto en un solo dispositivo.
- Agregar un reloj que diga el tiempo en el que ocurren las alarmas.

- Es recomendable programar un tiempo de hibernación de la red, según sea la clase de alarma más probable en el medio donde trabaje.
- Cambiar los módulos de comunicación actualmente utilizados por unos más baratos y que consuman menos energía.

Bibliografía

- [1] Navarro Murillo, Emmanuel, **Diseño e implementación de una red de comunicación inalámbrica para protección ambiental en el bosque**, instituto Tecnológico de Costa Rica, 2005
- [2] Tanenbaum, Andrew S, **Redes de Computadoras** 4ª Ed. Pearson, México, 2003
- [3] Cisco Systems, **Guía del primer año CCNA 1 y 2**, 3ª Ed. Pearson, España, 2004
- [4] Demirkol, I. **MAC Protocols for Wireless Sensor Networks** [en línea]: Network Research Laboratory, Computer Engineering Department of Bogazici University, Turkey.
<www.cmpe.boun.edu.tr/~ilker/IlkerDEMIRKOL_COMMAG.pdf>
- [5] Davis, W. B. **Graphical Model Theory for Wireless Sensor Networks** [en línea]: Lawrence Berkeley National Laboratory, University of California.
<<http://repositories.cdlib.org/cgi/viewcontent.cgi?article=2383&context=lbnl>>
- [6] Ye, W. **Medium Access Control in Wireless Sensor Networks** [en línea]: Information Sciences Institute, University of Southern California.
<www.isi.edu/~weiyu/pub/isi-tr-580.pdf>
- [7] Barton, J. **Miniaturized Modular Wireless Sensor Networks** [en línea]: National Microelectronic Research Centre Lee Maltings, Ireland.
<www.media.mit.edu/resenv/pubs/papers/2002-09-NMRC-UBICOMP-poster.pdf>
- [8] Chiasserini, C. F. **Modeling the Performance of Wireless Sensor Networks** [en línea]: Dipartimento di Elettronica, Politecnico di Torino.
<http://www.ieee-infocom.org/2004/Papers/06_1.PDF>
- [9] Lewis, F. L. **Wireless Sensor Networks** [en línea]: Automation and Robotics Research Institute, University of Texas at Arlington.
<<http://arri.uta.edu/acs/networks/WirelessSensorNetChap04.pdf>>.
- [10] Ganesan, Govindan, Shenker, Estrine **Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks** [en línea]: International computer science, University of Berkeley California. <<http://www.ietf.org/html.charters/manetcharter.html>>.

Apéndices:

Apéndice A1: Programa implementado en los nodos

```
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
////////////////////*Introducción al programa*////////////////////
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/*
    El diseño general del programa se hace de forma que las rutinas del protocolo se
    realicen enteramente en las subrutinas correspondiente a cada uno de los estados,
    mientras que el control del flujo se hace en el programa principal (main)
*/

/*Definición de pines del entrada / salida del pic *****/
//PIN_C7  RX      entrada  Tx del puerto serial
//PIN_C6  TX      salida   Rx del puerto serial
//PIN_B1  RTS     entrada  Señal de petición para transmitir
//PIN_B2  salida  Señal de control del led
//PIN_D4  CTS     salida  Señal de listo para recibir
//PIN_D5  In_range entrada  Indica cuando el nodo esta en el rango
//PIN_D6  C/D     salida  Controla la transmisión de comandos o datos
//PIN_D7  RESET   salida  Controla el reset del transceptor

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
////////////////////*programa*////////////////////
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#include <16F877A.h>
#include delay(clock=8000000)
#include HS,NOWDT
#include rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7)
#include NOWDT,NOPROTECT,HS,NOPUT,NOLVP,NOBROWNOUT,NOCPD,NOWRT
#define RAND_MAX 5
#include <STDLIB.H> //por la funcion rand hay que modificar la funcion para que el
rand_max=5

#include fast_io(A)
#include fast_io(B)
#include fast_io(D)

////////////////////*VARIABLES DEL PROGRAMA*////////////////////

#BYTE PORTA= 0x05
#BYTE PORTB= 0x06
#BYTE PORTD= 0x08

/*Variables *****/

INT cont=0; //Contador de datos recibidos*/
INT i; //Variable para contador*/
INT random; //almacena el valor aleatorio de espera del segundo estado de espera
INT x; //Variable para contador
```

```

BYTE dato;
BYTE tercerespera; //Contador del tercer estado de espera
BYTE segundoespera; //Contador del segundo estado de espera
BYTE mimac[3]; //Almacena la direccion mac del nodo
BYTE datos[6]; //Almacena los datos recibidos dinámicamente
BYTE info[6]; //Almacena los datos recibidos que se están utilizando
BYTE alarmstored[6]; /*Almacena si ahy una alarma en espera de ser retransmitida*/
BYTE nivel; /*Indica el nivel de pertenencia del nodo*/
BOOLEAN pt=FALSE; /*Indica que se recibió una trama PT válida*/
BOOLEAN rts=FALSE; /*Indica que se recibió una trama rts válida*/
BOOLEAN cts=FALSE; /*Indica que se recibió una trama cts válida*/
BOOLEAN alar=FALSE; /*Indica que se recibió una trama alar válida*/
BOOLEAN ack=FALSE; /*Indica que se recibió una trama ack válida*/
BOOLEAN rec=FALSE; /*Indica que se está recibiendo una trama válida*/
BOOLEAN trama=false; /*Indica que hay una trama a la que no se ha dado tratamiento*/
BOOLEAN esperarts=false; /*Indica que se está esperando una trama de RTS*/
BOOLEAN esperapt=false; /*Indica que se está esperando una trama pt */
BOOLEAN esperacts=false; /*Indica que se está esperando una trama de cts*/
BOOLEAN esperaalar=false; /*Indica que se está esperando una trama de alarma*/
BOOLEAN esperaack=false; /*Indica que se está esperando una trama de Acknowledge*/
BOOLEAN alarmpend=false; /*Indica que hay una alarma pendiente por transmitir*/
BOOLEAN falloespera=false; /*Indica cuando no se recibe CTS*/
BOOLEAN nocts=false; /*Indica que se recibió una trama CTS que no esta dirigida al nodo*/
BOOLEAN norecrts=false; /*Indica que no recibió una trama RTS*/
BOOLEAN norecalar=false; /*Indica que no recibió una trama de alarma*/
BOOLEAN norecpt=false; /*Indica que no recibió una trama de pt valida*/

```

```

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//////////////////////*Rutinas del programa*////////////////////////////////////
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

//Parpadeo del led para pruebas////////////////////////////////////
void parpadeo(){
    OUTPUT_LOW(PIN_B2);
    DELAY_ms(1000);
    OUTPUT_HIGH(PIN_B2);
    DELAY_ms(1000);
}
/*Reset por Hardware*****/
void reset_h(){
    OUTPUT_HIGH(PIN_D7);
    DELAY_ms(20);
    OUTPUT_LOW(PIN_D7);
    DELAY_ms(10);
}
/*Envio de alarma pendiente*****/
VOID env_alarmpend(){
    FOR(i=0;i<6;++i){
        PUTC(alarmstored[i]);
    }
}

```

```

//habilita recibir datos broadcast *****/
VOID mem56h(){
    OUTPUT_LOW(PIN_D6);    //Activa recepción de comandos
    DELAY_ms(2);
    PUTC(0xCC);           //Inicia secuencia de comandos
    DELAY_ms(1);
    PUTC(0xc1);           //Indica comando a realizar
    DELAY_ms(1);
    PUTC(0x56);           //Envía la dirección
    DELAY_ms(2);
    PUTC(0x01);           //Envía la cantidad de datos a enviar
    DELAY_ms(2);
    PUTC(0x43);           //Envía el dato a enviar
    DELAY_ms(3);
    OUTPUT_HIGH(PIN_D6);  //Activa recepción de datos
    DELAY_ms(300);}

//Configura el transceptor en modo broadcast*****/
VOID broadcast(){
    OUTPUT_LOW(PIN_D6);    //Activa recepción de comandos
    DELAY_ms(2);
    PUTC(0xCC);           //Inicia secuencia de comandos
    DELAY_ms(1);
    PUTC(0x08);           //Indica comando a realizar
    DELAY_ms(1);
    PUTC(0x01);           //Envía primer byte de la dirección
    DELAY_ms(2);
    OUTPUT_HIGH(PIN_D6);  //Activa recepción de datos
    DELAY_ms(300);
}

//Pone el transceptor en sueño profundo *****/
VOID deepsleep(){
    OUTPUT_LOW(PIN_D6);    //Activa recepción de comandos
    DELAY_ms(2);
    PUTC(0xCC);           //Inicia secuencia de comandos
    DELAY_ms(1);
    PUTC(0x86);           //Indica comando a realizar
    DELAY_ms(300);
    OUTPUT_HIGH(PIN_D6);  //Activa recepción de comandos
    DELAY_ms(300);
}

//Envío de trama PT*****/
void envpt(){
    putc(0xF1);
    putc(nivel);
    putc(mimac[0]);
    putc(mimac[1]);
    putc(mimac[2]);
    putc(0xF1);
}

//Envío de trama CTS*****/
void envcts(){

```

```

putc(0xF3);
putc(nivel);
putc(datos[2]);
putc(datos[3]);
putc(datos[4]);
putc(0xF3);
}
//Envio de trama ACK*****
void envack(){
putc(0xF5);
putc(alarmstored[1]);
putc(alarmstored[2]);
putc(alarmstored[3]);
putc(alarmstored[4]);
putc(alarmstored[5]);
}

```

```

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//////////////////////*Estados del programa*//////////////////////
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/*Inicializacion de los componentes del nodo*
/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

void inic_comp_nodo(){
IF(BIT_TEST(PORTD,5)){
reset_h();
delay_ms(5000);
}
nivel=255;
tercerespera=0;
segundoespera=0;
datos[1]=0;
mimac[0]=0X11;
mimac[1]=0XA3;
mimac[2]=0X3C;
mem56h();
broadcast();
cont=0;
rec=false;
}

```

```

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/*Descubrimiento del nivel de pertenencia*
/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

void desc_niv_pert(){
esperapt=true;

```

```

delay_ms(5600);
esperapt=false;
pt=false;

alarmstored[0]=0xF4;
alarmstored[1]=0x00;
alarmstored[2]=nivel;
alarmstored[3]=mimac[0];
alarmstored[4]=mimac[1];
alarmstored[5]=mimac[2];
alarmpend=true;
}

```

```

/*%%%%%%%%%%*/
/*Estado de Hibernación*/
/*%%%%%%%%%%*/

```

```

void hibernacion(){
deepsleep();
delay_ms(2800);
tercerespera=0;
segundoespera=0;
/*
If(alarmapend==false){
if(bit_test(pota,2){ //El pin A2 tiene la entrada de fuego
alarmstored[0]=0x04;
alarmstored[1]=0x03;
alarmstored[2]=nivel;
alarmstored[3]=mimac[0];
alarmstored[4]=mimac[1];
alarmstored[5]=mimac[2];
alarmpend=true;
output_high(pin_a3); //El pin A3 tiene la senal de recibido de fuego
goto listo;
}else{
if(bit_test(pota,4){ //El pin A4 tiene la entrada de Motosierras
alarmstored[0]=0x04;
alarmstored[1]=0x02;
alarmstored[2]=nivel;
alarmstored[3]=mimac[0];
alarmstored[4]=mimac[1];
alarmstored[5]=mimac[2];
alarmpend=true;
output_high(pin_a5); //El pin A5 tiene la senal de recibido de motosierras
goto listo;
}else{
if(bit_test(pota,6){ //El pin A6 tiene la entrada de disparos
alarmstored[0]=0x04;
alarmstored[1]=0x01;
alarmstored[2]=nivel;
alarmstored[3]=mimac[0];
alarmstored[4]=mimac[1];
alarmstored[5]=mimac[2];
*/

```

```

alarmpend=true;
output_high(pin_a7); //El pin A7 tiene la senal de recibido de disparos
}}
listo:*/
reset_h();
delay_ms(2000);
}

```

```

/*%%%%%%%%%%%%%%*/
/*Primer estado de espera*/
/*%%%%%%%%%%%%%%*/

```

```

void primerestadodeespera(){
i=0;
esperapt=true;
ciclo:
if((rec==false)&&(pt)){
esperapt=false;
pt=false;
}else{
i=i+1;
delay_ms(2);
if(i<2800){
goto ciclo;
}else{
norecpt=true;
nivel=nivel+1;
esperapt=false;
}
}
}
}
}

```

```

/*%%%%%%%%%%%%%%*/
/*Segundo estado de espera*/
/*%%%%%%%%%%%%%%*/

```

```

void segundoestadodeespera(){
random=rand();
esperacts=true;
for(i=0;i<random;i++){
delay_ms(350);
}
if(nocts){
segundoespera= segundoespera+1;
esperacts= false;
nocts=false;
cts=false;
}
else{
esperacts=false;
cts=false;
segundoespera=0;
PUTC(0xF2);
putc(nivel);
putc(mimac[0]);
}
}
}
}
}

```

```

putc(mimac[1]);
putc(mimac[2]);
putc(0xF2);
}

```

```

/*%%%%%%%%%%%%%%*/
/*Tercer estado de espera*/
/*%%%%%%%%%%%%%%*/

```

```

void tercerestadodeespera(){
i=0;
esperacts=true;
ciclo:
if((cts)&&(rec==false)){
goto fin;
}else{
delay_ms(2);
if(i<=200){
i=i+1;
goto ciclo;
}else{
goto fallo;
}}
fallo:
falloespera=true;
esperacts=false;
tercerespera=tercerespera+1;
fin:
cts=false;
esperacts=false;
}

```

```

// Define el tiempo de espera del tercer estado de espera

```

```

/*%%%%%%%%%%%%%%*/
/*Cuarto estado de espera*/
/*%%%%%%%%%%%%%%*/

```

```

void cuartoestadodeespera(){
inicio:
i=0;
env_alarmpend();
ciclo:
esperaack=true;
if((ack)&&(rec==false)){
if((alarmstored[1]==datos[1])&&(alarmstored[2]==datos[2])&&(alarmstored[3]==datos[3])&&(alarmstore
d[4]==datos[4])&&(alarmstored[5]==datos[5])){
alarmpend=false;
ack=false;
esperaack=false;
}
else{
ack=false;
goto inicio;}
}else{
delay_ms(2);
if(i<=200){
i=i+1;
goto ciclo;
}
}
}

```

```

// Define el tiempo de espera del cuarto estado de espera

```

```

}else{
esperaack=false;
ack=false;
esperaack=false;
}}}

```

```

/*%%%%%%%%%%%%%%*/
/*Primera fase de petición de trama*/
/*%%%%%%%%%%%%%%*/

```

```

void primpettrama(){
i=0;
envpt();
esperarts=true;
ciclo:
if((rts)&&(rec==false)){
rts=false;
esperarts=false;
norecrts=false;
}else{
delay_ms(2);
if(i<=200){ // Define el tiempo de espera para recibir un rts
i=i+1;
goto ciclo;
}else{
norecrts=true;
esperarts=false;
rts=false;
}}}

```

```

/*%%%%%%%%%%%%%%*/
/*Segunda fase de petición de trama*/
/*%%%%%%%%%%%%%%*/

```

```

void segpettrama(){
envcts();
}

```

```

/*%%%%%%%%%%%%%%*/
/*Escucha*/
/*%%%%%%%%%%%%%%*/

```

```

void escucha(){
i=0;
esperaalar=true;
ciclo:
if((alar)&&(rec==false)){
esperaalar=false;
alarmstored[0]=0xF4;
alarmstored[1]=datos[1];
alarmstored[2]=datos[2];
alarmstored[3]=datos[3];
alarmstored[4]=datos[4];
alarmstored[5]=datos[5];
alar=false;
norecalar=false;
}else{
delay_ms(2);
}
}

```



```

        nivel=datos[1]+1;}

        //////////////////////////////////
}ELSE IF(dato==0xF2){          // *trama RTS* //
    rec=true;                  //////////////////////////////////
    if((datos[1]>nivel)&&(esperarts)){
        rts=true;}

        //////////////////////////////////
}ELSE IF(dato==0xF3){          // *Trama CTS* //
    rec=true;                  //////////////////////////////////
    if(esperacts){
        if((datos[2]==mimac[0])&&(datos[3]==mimac[1])&&(datos[4]==mimac[2])){
            cts=true;}
        else{
            nocts=true;}
        }

        //////////////////////////////////
}ELSE IF(dato==0xF4){          // *trama Alarma* //
    rec=true;                  //////////////////////////////////
    if(esperaalar){
        alar=true;}

        //////////////////////////////////
}ELSE IF(dato==0xF5){          // *trama Ack* //
    rec=true;                  //////////////////////////////////
    if(esperaack){
        ack=true;}
}ELSE{
    rec=false;
}

IF(rec){
    cont=cont+1;
    IF(cont==6){
        rec=false;
        cont=0;
    }
}
OUTPUT_LOW(PIN_D4);           //Activa la señal de listo para recibir

} //end int_rda

```

```

//%%%%%%%%%%
//////////////////*Programa Principal*//////////////////
//%%%%%%%%%%

```

```

void main(){
//Configuración del microcontrolador
  setup_adc_ports(NO_ANALOGS);
  setup_adc(ADC_CLOCK_DIV_2);
  setup_spi(FALSE);
  setup_psp(PSP_DISABLED);
  setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_ccp1(CCP_OFF);
  setup_ccp2(CCP_OFF);
  enable_interrupts(INT_RDA);
  enable_interrupts(global);

  SET_TRIS_A(0x00);
  SET_TRIS_B(0x02);
  SET_TRIS_D(0x20);

//Inicialización del pic////////////////////////////////////

  RESET_H();
  OUTPUT_LOW(PIN_D7);
  OUTPUT_HIGH(PIN_D6);
  OUTPUT_LOW(PIN_D4);
  OUTPUT_LOW(PIN_A0);
  BIT_SET(PORTB,2);
  DELAY_ms(2000);

//Ciclo principal y control de flujo////////////////////////////////////
  WHILE (TRUE) {
  parpadeo();           //Estado
  delay_ms(2000);
  inic_comp_nodo();     //Estado
  desc_niv_pert();     //Estado
  goto primestesp;

  hiber:
  hibernacion();       //Estado
  if (alarpnd){
  goto primestesp;
  }else{
  goto primfaspettram;
  }
  primestesp:
  primerestadodeespera(); //Estado
  if(norecpt){
  norecpt=false;
  goto hiber;
  }
}

```

```

segundoestadodeespera();          //Estado

switch(segundoespera){
case 0: goto terestesp;
case 1: goto primestesp;
case 2: goto primestesp;
case 3: goto hiber;
default: goto hiber;
}

terestesp:
tercerestadodeespera();          //Estado
if (falloespera){
falloespera=false;
if (tercerespera>=3){
goto hiber;}
else{
goto primestesp;
}}

cuartoestadodeespera();          //Estado
goto hiber;

primfaspettram:
primpettrama();
if (norecrts){
norecrts=false;
goto hiber;
}
segpettrama();
escucha();
if (norecalar){
goto hiber;
}
verificacion();
goto primestesp;

}}

```

Apéndice A2: Antecedentes prácticos:

Anteriormente se desarrollo un trabajo sobre el protocolo de la “capa de red”, el cual podría tomarse como base con algunas modificaciones para el desarrollo del protocolo de la capa de red y enlace de datos en el presente proyecto. Este trabajo anteriormente mencionado fue asesorado por el profesor Ing Néstor Hernández, realizado por Emmanuel Navarro Murillo, y documentado en su informe de proyecto de graduación “Diseño e implementación de una red de comunicación inalámbrica para protección ambiental en el bosque” presentado en julio del año 2005 en el Instituto tecnológico de Costa Rica.

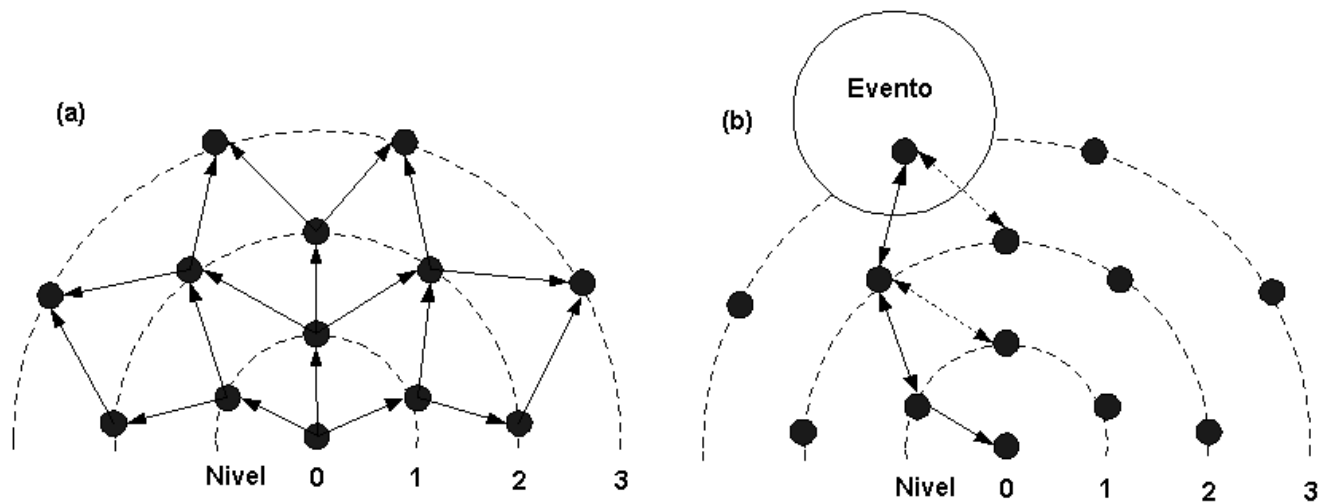


Figura A2. 1: Descripción de la topología seleccionada

En este trabajo inicialmente se conciben niveles lógicos en la red, como se muestra en la figura A2.1, la distancia entre las capas de la red estará definida por el rango de alcance de los futuros telé detectores, así las capas empezarán en la estación base con nivel cero y se irán incrementando conforme se aumente la distancia. En cada nivel los nodos estarán numerados, lo cual permitirá reconocer de forma sencilla a sus vecinos. En este entendido, cuando se detecte un evento, los nodos sólo se comunicarán con sus vecinos del mismo nivel o con los pertenecientes al nivel

inferior. Con esto los nodos no necesitan saber su ubicación geográfica, pues con su identificación de red conocen cuales nodos les son más cercanos

El protocolo de red se implementa de la siguiente manera:

Una vez que se detecta una alarma²³ se solicita el estado de los dos nodos inferiores; si alguno de los nodos inferiores, o los dos nodos inferiores no contestan entonces se procede a pedir el estado de uno o los dos nodos vecinos como se muestra en los trozos de código siguientes²⁴

```
/******Rutina para atender una alarma******/  
VOID ALARMA(){25  
  
    UPDATED=FALSE;26 → // Esta variable indica que se actualizó la  
  
                                //información de los nodos para transmitir  
  
    REENVI[0]=ESTADO[0]; → // Se definen los datos de estado del nodo para  
  
    REENVI[1]=ESTADO[1]; // ser eventualmente enviados por un ask retrans  
    REENVI[2]=ESTADO[5];  
    REENVI[3]=ESTADO[6];  
  
    REQUEST(); → // Se llama a la rutina request  
  
}
```

²³ Aún no se ha definido la manera en que las alarmas son detectadas debido a que no se tienen aún los telé detectores.

²⁴ Este código al igual que los siguientes segmentos de código que se muestran en el presente informe se programaron en "PIC C" , con el objetivo de ser implementados sobre un pic 16f874a de Microchip

²⁵ Se debe notar que este método no se llama como parte del código principal ya que no se ha definido la manera en la que se detectarán las alarmas.

²⁶ Esta línea es redundante porque se repite en la rutina "request" la cual se llama a continuación

/****** Solicita el estado de los nodos vecinos y decide a cual enviar *****/

```
VOID REQUEST(){
```

```
  UPDATED=FALSE;
```

```
  WHILE(!UPDATED){ → // la variable updated se actualiza en la rutina
```

```
                    // Status_Req una vez que se ha recibido el estado  
                    //de un nodo
```

```
  CARGA=0x00; → //inicializa la variable carga que se utiliza en la rutina
```

```
                    //Status_Req para comparar y encontrar el nodo con  
                    // mayor carga
```

```
  x=0; → //Esta variable se utiliza para saber cuantos nodos se
```

```
        // han consultado y además han respondido correctamente con  
        // su información de estado
```

```
  STATUS_REQ (INF01); → //Pide la información de estado del nodo "INF01"
```

```
  STATUS_REQ (INF02); → //Pide la información de estado del nodo "INF01"
```

```
  IF(x<2){ → //si los nodos que respondieron correctamente son menos de
```

dos.

```
  STATUS_REQ (VEC01); → //Pide la información de estado del nodo "VEC01"
```

```
}
```

```
IF(x<2){ → //si los nodos que respondieron correctamente son menos de dos.
```

```
STATUS_REQ (VEC02); → //Pide la información de estado del nodo "VEC01"
```

```
}
```

```
IF(!UPDATED){ → //En caso de que ningún nodo haya contestado.
```

```
DELAY_ms(150000); → //espera 150 segundos para reiniciar el ciclo
```

```
}}
```

```
ASK_RETRANS(); → // Solicita que se retransmita al nodo que tenga mayor
```

```
//carga cuya dirección se guardó en la rutina "Status_Req"
```

```
UPDATED=FALSE; → //Inicializa la variable "updated"
```

```
DELAY_ms(20);}
```

El resto de las rutinas involucradas se pueden revisar en el anexo1 donde se observa el código completo desarrollado por Emmanuel.

Una vez que se tiene el nodo de mayor carga se procede a enviar a este la solicitud de transmitir con una trama como la que se describe a continuación²⁷

²⁷ Se puede ver de la figura que el byte "capa del nodo emisor de alarma" es redundante ya que si se cuenta con la identificación de red del nodo emisor de la alarma ya se conoce la capa de donde proviene.

0x33	capa del nodo emisor de alarma	Identificación de red del nodo emisor de alarma	estado de batería del nodo emisor de alarma	Alarma	0x33
------	--------------------------------	---	---	--------	------

Figura A2. 2: Ilustración de las tramas de solicitud de retransmisión

Este procedimiento se repite hasta llegar al nodo de nivel cero en el que el dato debe ser procesado por un software y mostrado al guarda parques para que la alarma pueda ser respondida a tiempo.

A2.1 Mejoras propuestas al trabajo realizado:

La idea original que se plantea para buscar la ruta del mensaje de acuerdo con el nivel de carga de los nodos, (protocolo de la capa de red) resulta sumamente útil ya que se adapta de una manera excelente a los requerimientos del problema propuesto inicialmente con base al esquema utilizado en el que los nodos se encuentran permanentemente encendidos.

No obstante, en el código original se deja abierta la posibilidad de que el mensaje viaje de un nodo al vecino en el caso de que este cuente con una carga mayor que la de uno de los nodos inferiores, y que el otro nodo inferior no conteste; esto podría darse no solamente por que el nodo que no contestó no tenga carga, sino que podría darse por efectos de desvanecimiento, corrupción de la trama enviada, o de la recibida. Si se considera un caso como el que plantea en la figura 2.3 donde se puede ver en azul la carga de

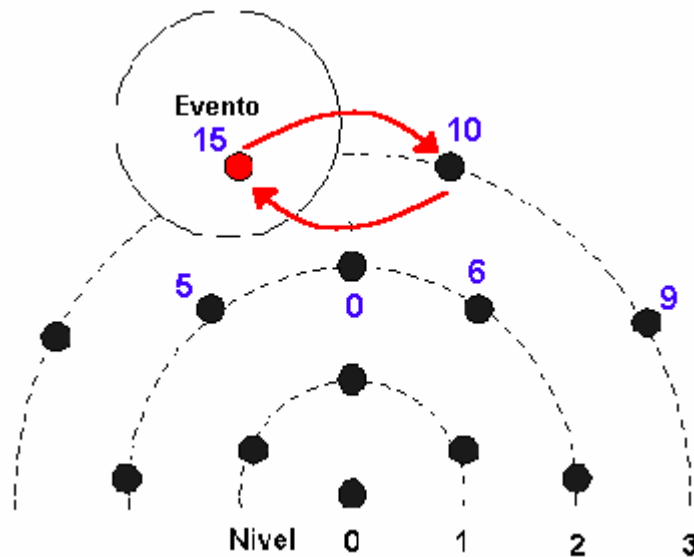


Figura A2.3: Ejemplo de error de ciclo en el protocolo de red

cada uno de los nodos, y se resalta en rojo el nodo en el que se produce una alarma; se puede ver que con el protocolo de red descrito inicialmente se produciría un ciclo en el que el mensaje de retransmisión se enviaría entre dos nodos vecinos hasta descargarlos a un nivel inferior que el de sus nodos inferiores lo cual no solamente sería un desperdicio de energía sino que además retrasa la atención de la alarma.

Para evitar esta situación se plantea que los nodos no puedan retransmitir un mensaje al nodo del cual lo recibieron.

Para esto es necesario incluir en la trama de petición de retransmisión el identificador de red del ultimo nodo que la transmitió. Tomando en cuenta, como se apuntó anteriormente, que el campo ocupado por la capa del nodo emisor de la alarma es redundante se puede eliminar de forma que la nueva trama quedaría de la siguiente manera:

0x33	Identificacion de red del ultimo nodo emisor	Identificacion de red del nodo emisor de alarma	estado de bateria del nodo emisor de alarma	Alarma	0x33
------	--	---	---	--------	------

Figura A2.4: Trama de petición de retransmisión modificada

Además se debe modificar el código de la rutina REQUEST() como se muestra a continuación

```
VOID REQUEST(){
UPDATED=FALSE;
WHILE(!UPDATED){
    CARGA=0x00;
    x=0;
    STATUS_REQ (INF01);
    STATUS_REQ (INF02);

    IF((x<2) &&(VEC01[1] != REENVI[0])){ → //si los nodos que respondieron
                                                //correctamente son menos de
                                                //dos y además el vecino 1 no
                                                //envió la petición de reenvío

        STATUS_REQ (VEC01);
    }

    IF((x<2) &&(VEC02[1] != REENVI[0])){ → //si los nodos que respondieron
                                                //correctamente son menos de
                                                //dos y además el vecino 2 no
                                                //envió la petición de reenvío

        STATUS_REQ (VEC02);
    }
    IF(!UPDATED){
        DELAY_ms(150000);
    }
}
ASK_RETRANS();
```

```

UPDATED=FALSE;
DELAY_ms(20);
}

```

Además se deben modificar las rutinas Alarma() y Retrans() de la siguiente forma:

```

VOID ALARMA(){

```

```

    REENVI[0]=ESTADO[1]; → // se usa el identificador de red del nodo que

```

```

        //envía la petición.

```

```

    REENVI[1]=ESTADO[1];

```

```

    REENVI[2]=ESTADO[5];

```

```

    REENVI[3]=ESTADO[6];

```

```

    REQUEST();

```

```

}

```

```

VOID RETRANS(){

```

```

    UPDATED=FALSE;

```

```

    REENVI[0]=ESTADO[1]; → // se usa el identificador de red del nodo que

```

```

        // reenvía la petición.

```

```

    REENVI[1]=DATOS[2];

```

```

    REENVI[2]= DATOS[3];

```

```

    REENVI[3]= DATOS[4];

```

```

    REQUEST(); }

```

```

    IF(ESTADO[0]>1){

```

```

        REQUEST();

```

```

    }ELSE{

```

```

        ASK_RETRANS();

```

```

    }}

```

Apéndice A3: Suma de variables aleatorias

Sean dos variables estocásticas X_1 y X_2 con probabilidad conjunta $P_X(x_1; x_2)$: La probabilidad de que la variable $Y = X_1 + X_2$ tenga un valor entre y y $y + \Delta y$ es:

$$P_Y(y)\Delta y = \iint_{y < x_1 + x_2 < y + \Delta y} P_X(x_1, x_2) dx_1 dx_2$$

que es equivalente a:

$$\begin{aligned} P_Y(y) &= \iint \delta(x_1 + x_2 - y) P_X(x_1, x_2) dx_1 dx_2 \\ &= \int P_X(x_1, y - x_1) dx_1 \end{aligned}$$

Si X_1 y X_2 son independientes entonces

$$P_Y(y) = \int P_{X_1}(x_1) P_{X_2}(y - x_1) dx_1,$$

esto es la convolución de las dos probabilidades individuales.