



Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

**Protocolo de encaminamiento de datos
por TDMA para una red de sensores
inalámbrica con topología estructurada
para un ambiente ruidoso**

Idónea Comunicación de Resultados
que para obtener el grado de:
**MAESTRO EN CIENCIAS Y
TECNOLOGÍAS DE LA INFORMACIÓN**
presenta
Ing. Néstor López Montiel

Asesor:
Dr. Luis Martín Rojas Cárdenas

Defendida públicamente en la UAM-Iztapalapa frente al jurado integrado por:

Dra. Martha María Montes de Oca Cáliz - Universidad Nacional Autónoma de México

Dr. Miguel López Guerrero - Universidad Autónoma Metropolitana

Dr. Luis Martín Rojas Cárdenas - Universidad Autónoma Metropolitana

5 de septiembre de 2014



Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

**Protocolo de encaminamiento de datos
por TDMA para una red de sensores
inalámbrica con topología estructurada
para un ambiente ruidoso**

Idónea Comunicación de Resultados
que para obtener el grado de:
**MAESTRO EN CIENCIAS Y
TECNOLOGÍAS DE LA INFORMACIÓN**
presenta
Ing. Néstor López Montiel

Asesor:
Dr. Luis Martín Rojas Cárdenas

Defendida públicamente en la UAM-Iztapalapa frente al jurado integrado por:

Dra. Martha María Montes de Oca Cáliz - Universidad Nacional Autónoma de México

Dr. Miguel López Guerrero - Universidad Autónoma Metropolitana

Dr. Luis Martín Rojas Cárdenas - Universidad Autónoma Metropolitana

5 de septiembre de 2014

Agradecimientos

Con todo mi cariño y amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando lo necesitaba, a ustedes por siempre mi corazón y mi agradecimiento.

Papá, Mamá, Llita, Luis y Emmanuel

A tu paciencia y comprensión, por sacrificar tu tiempo para ayudarme a cumplir con mis sueños. Por tu bondad y cariño me inspiraste a ser mejor una mejor persona para tí, ahora puedo decir que esta tesis también es tuya, gracias por estar siempre a mi lado.
Azalia.

A todos mis amigos que siempre estuvieron ahí en los momentos difíciles de mi vida. Por sus consejos y regaños.

A mi asesor Dr. Luis Rojas Cárdenas, por apoyarme y guiarme en mi vida académica.

Resumen

Las Redes de Sensores Inalámbricas (RSI) son ampliamente utilizadas en sistemas para la adquisición de datos. Bajo este esquema, los sensores capturan y transmiten pequeñas cantidades de datos en periodos regulares de tiempo. Las RSI usan protocolos de comunicación como IEEE 802.15.4, Zigbee entre otros. Estos protocolos, generalmente, no son compatibles con los protocolos de la pila TCP/IP [19].

Para las redes de sensores, es deseable contar con un soporte protocolario dinámico que permita integrarse y abandonar la red en cualquier momento, además, que ofrezca la autoconfiguración del sensor y la búsqueda de servicios. En cuanto a la topología de la red, ésta determina en gran medida las características de un protocolo para el encaminamiento y el transporte de datos. En este aspecto, se puede hablar de dos tipos de redes: redes ad-hoc y redes fijas. En las redes ad-hoc, los sensores se organizan por sí mismos en clústeres, los cuales se intercomunican gracias a funcionalidades de encaminamiento. En cambio, en una red fija, un elemento con funcionalidades adicionales, conocido como punto de acceso maneja las comunicaciones internas en el grupo y hacia el exterior con otros grupos de sensores.

En este trabajo de investigación, se aborda el diseño de una arquitectura para realizar la adquisición de datos en una red de sensores con topología lineal estática, la cual tiene como particularidades adicionales la capacidad de trabajar en un ambiente altamente ruidoso. Se estudiaron diferentes tecnologías existentes, con el objetivo de encontrar aquella que nos ofreciera la mejor solución a la problemática de este trabajo de investigación. Se analizaron las ventajas y desventajas de tecnologías como Zigbee, MiWi y Bus CAN. Éstas nos ofrecen métodos de acceso al medio, así como técnicas de autoconfiguración y control de errores, sin embargo, no todas están hechas para trabajar en ambientes ruidosos, el cual es el objetivo de este trabajo de investigación.

Con este trabajo se logró que una red de sensores inalámbrica pueda operar de manera eficiente en ambientes con ruido. Para ello se diseñó un protocolo de acceso al medio, basado en TDMA, y un protocolo de encaminamiento de paquetes, que no depende de los mensajes

de acuse, para reducir los efectos del ruido en el proceso de comunicación. Los resultados de la simulación demuestran que es posible garantizar un tiempo máximo para la entrega de paquetes que van de los sensores a la estación base.

Abstract

WSNs (Wireless Sensor Networks) are widely used in systems for data acquisition. Under this scheme, sensors acquire and transmit data streams that often have a fixed pattern: they transmit small amounts of data at regular intervals of time. Protocols such as TCP / IP are discarded in this context due to their complexity and resource requirements.

The topology of a sensor network determines the characteristics of the routing protocol and medium access used. In this regard, one can speak of two types of networks: ad-hoc networks and static networks. In ad-hoc networks, sensors organize themselves into clusters, which intercommunicate through a routing functionality. Instead, in a static network, an element with additional functionality, known as access point, handles internal communications in the group and out of it with other groups of sensors.

In this research, we address the design of an architecture for data acquisition in a sensor network with a static linear topology, which has as an additional characteristic that it has the ability to work in a highly noisy environment. Different existing technologies were studied in order to find one that offered us the best solution for this research problem. The advantages and disadvantages of technologies such as ZigBee, MiWi and CAN Bus were analyzed. These methods provide medium access, autoconfiguration techniques and error control, however, not all of them are made to work in noisy environments, which is the aim of this research.

This goal was achieved by designing a medium access protocol, based on Time Division Multiple Access (TDMA), and a packet routing protocol, which is not dependent on acknowledgment messages, to reduce the effects of noise in the communication process. The simulation results show that it is possible to guarantee a maximum time for delivery of packets transmitted from the sensor nodes to the base station.

Acrónimos

AODV	Ad hoc On-Demand Distance Vector
ARQ	Automatic Repeat Request
CSH	C Shell
EMT	Encaminamiento por Multiplexión de Tiempo
FEC	Forward Error Correction
FPGA	Field Programmable Gate Array
GPL	General Public License
HARQ	Hibryd Automatic Repeat Request
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
NACK	Negative Acknowledgement
NS	Network Simulator
PDA	Personal Digital Assistant
QoS	Quality of Sevice
RAM	Random Access Memory
RSI	Redes de Sensores Inalámbricas
SCR	Silicon Controlled Rectifier
TCL	Tool Command Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UPS	Uninterruptible Power Supply
WSN	Wireless Sensor Network

Contenido

Lista de figuras	XIII
Lista de tablas	XV
1. Introducción	1
1.1. Motivación	1
1.2. Estructura del trabajo de investigación	2
2. Redes de sensores inalámbricas	3
2.1. Arquitectura del sistema	4
2.1.1. Nodo inalámbrico	4
2.1.2. Puerta de enlace	6
2.1.3. Estación base	7
2.2. Protocolos de encaminamiento de los datos	7
2.2.1. Clasificación de las redes de sensores inalámbricas	7
2.2.2. Diseño de un protocolo de encaminamiento para redes de sensores	8

2.2.3. Protocolos de encaminamiento	9
2.3. El control de acceso al medio	10
2.4. Mecanismos para el control de errores	11
2.5. Redes de sensores inalámbricas y el ruido	12
3. Protocolo propuesto	15
3.1. Topología de la red de sensores	16
3.2. Protocolo de acceso al medio	17
3.3. Protocolo de encaminamiento de datos	18
4. Evaluación de desempeño	21
4.1. Simulación de una red inalámbrica de sensores	21
4.1.1. El simulador de redes NS-2	21
4.1.2. Simulación del protocolo de encaminamiento y acceso al medio	22
4.1.3. Escenario de simulación	22
4.1.4. Parámetros de simulación	23
4.1.5. Script de simulación tcl	25
4.2. Implementación de la red de nodos sensores	25
5. Resultados obtenidos en la simulación y en la red prototipo	31
5.1. Resultados de la simulación	31
5.2. Porcentaje de pérdidas	32

CONTENIDO	XI
5.3. Tiempo de entrega de paquetes	36
5.4. Pruebas con la implementación de la red de sensores	39
6. Conclusiones y recomendaciones para trabajo futuro	41
Referencias	43
Apéndice	45
A. Script TCL	47
B. Scripts de filtrado de datos	51
C. Diagrama del circuito sensor	57

Lista de figuras

2.1. Red de sensores inalámbrica.	5
2.2. Arquitectura de un nodo sensor inalámbrico.	5
3.1. Topología - red de sensores inalámbrica.	17
3.2. Control de acceso al medio.	18
3.3. Inteferencia casusada por los nodos de la red.	20
3.4. Encaminamiento de los datos.	20
4.1. Escenario de simulación.	23
4.2. Formato del mensaje de datos UDP.	24
4.3. Ejemplo de codificación Manchester.	26
4.4. Trama del mensaje en los circuitos de prueba.	27
4.5. Autómata de operación de la estación base.	29
4.6. Autómata de la operación de los nodos.	29
4.7. Red de sensores de prueba.	30
5.1. Distancias de los nodos con respecto a la estación base	32

5.2. Porcentaje de pérdidas escenario 1. Encaminamiento usando AODV, con un rango de transmisión de 15 m.	33
5.3. Porcentaje de pérdidas escenario 3. Encaminamiento usando EMT, con un rango de transmisión de 15 m.	34
5.4. Porcentaje de pérdidas escenario 2. Encaminamiento usando AODV, con un rango de transmisión de 40 m.	35
5.5. Porcentaje de pérdidas escenario 4. Encaminamiento usando EMT, con un rango de transmisión de 40 m.	35
5.6. Tiempo de entrega de paquetes escenario 1. Encaminamiento usando AODV, con un rango de transmisión de 15 m.	36
5.7. Tiempo de entrega de paquetes escenario 3. Encaminamiento usando EMT, con un rango de transmisión de 15 m.	37
5.8. Tiempo de entrega de paquetes escenario 2. Encaminamiento usando AODV, con un rango de transmisión de 40 m.	38
5.9. Tiempo de entrega de paquetes escenario 4. Encaminamiento usando EMT, con un rango de transmisión de 40 m.	38
C.1. Diagrama esquemático del sensor	57

Lista de tablas

2.1. Fuentes de interferencia (basado en [18].)	13
4.1. Parámetros de simulación.	23
4.2. Valores utilizados para la codificación de nibbles	27
5.1. Escenarios de simulación	32

Lista de algoritmos

1. Algoritmo utilizado para inducir pérdidas de paquetes 25

Introducción

En los últimos años el uso de las redes de sensores inalámbricas ha crecido exponencialmente, integrándose en una gran cantidad de áreas de aplicación. Esto debido al bajo costo de despliegue, el bajo consumo de energía y el tamaño reducido de los nodos sensores.

Una RSI (red de sensores inalámbrica) es un conjunto de dispositivos capaces de medir las condiciones del ambiente a través de diversos sensores. Estos dispositivos tienen la capacidad de recibir y enviar información de forma inalámbrica a uno o más dispositivos a la vez, tales como ordenadores portátiles, PDA (*Personal Digital Assistant* - Asistente Digital Personal), teléfonos móviles y, principalmente, pequeños dispositivos equipados con las características básicas para la recepción o monitorización de un evento, el almacenamiento y posterior procesamiento de la información. En estas redes típicamente se realiza el envío de la información a una estación base con una mayor capacidad de procesamiento de información. Este envío normalmente requiere que la información sea retransmitida por los nodos intermedios, mediante varios saltos hasta alcanzar la estación base, la cual puede ser fija o estar en movimiento.

1.1. Motivación

Las redes de sensores inalámbricas permiten el desarrollo e implementación de una amplia gama de aplicaciones relacionadas con la domótica, medio ambiente, seguridad, salud, entornos industriales y monitorización [1, 22, 23, 24, 25, 26, 27]. Sin embargo, existen todavía algunas áreas de aplicación que no han sido abordadas de forma eficiente y en su totalidad, entre ellas, las redes de sensores inalámbricas con capacidad de operar en ambientes ruidosos. Algunas propuestas orientadas a este tipo de aplicaciones no disponen de protocolos de en-

caminamiento específicos y, muchas veces, hacen uso de protocolos de encaminamiento para redes *ad hoc* y no para RSI, ofreciendo prestaciones que no son plenamente eficientes.

De forma general, a continuación se presentan los principales retos de este trabajo de investigación:

- Diseñar un protocolo de acceso al medio, capaz de minimizar la pérdida de paquetes por colisiones en ambientes ruidosos.
- Diseñar un protocolo de encaminamiento para redes de sensores inalámbricas capaz de operar en ambientes ruidosos.
- Detectar errores en la recepción de paquetes en una red de sensores inalámbrica en ambientes ruidosos.
- Implementar con los protocolos diseñados una red de sensores inalámbrica, con una calidad de servicio, con la que se pueda garantizar la entrega de al menos un paquete de cada nodo de la red, en un tiempo máximo.

1.2. Estructura del trabajo de investigación

Este trabajo está estructurado de la siguiente manera: en el capítulo 2 se presenta un resumen de las principales características de las redes de sensores inalámbricas, su funcionamiento y principales componentes, además se hace un breve estudio de los protocolos de encaminamiento de datos y de acceso al medio, así como los mecanismos de detección y corrección de errores usados en redes de sensores. En el capítulo 3 se describe la propuesta, con la cual se logra implementar una red de sensores con la capacidad de operar en un ambiente ruidoso, lo que ocasiona la pérdida de paquetes; se presentan las propuestas de los protocolos de encaminamiento y acceso al medio usadas para conseguir el objetivo de este trabajo de investigación. En el capítulo 4 se muestra la manera con la cual evaluamos los protocolos propuestos, así como los escenarios y parámetros utilizados en las simulaciones hechas en este trabajo. Los resultados obtenidos de estas simulaciones se presentan en el capítulo 5. Más específicamente, aquí se presenta la explicación y justificación de los resultados obtenidos.

Por último se presenta un breve resumen del trabajo realizado y de los resultados obtenidos, así como la conclusión con algunas recomendaciones para trabajo futuro.

Redes de sensores inalámbricas

El avance tecnológico ha permitido que existan dispositivos pequeños, móviles o fijos, de bajo costo económico, capaces de monitorear diversos fenómenos como son: temperatura, humedad, presión, entre otros. La información adquirida por estos dispositivos es tratada y almacenada en una estación base central. A esto se le conoce como red de sensores inalámbrica.

La monitorización de los eventos en una RSI, se realiza en el dispositivo llamado nodo sensor, éstos están constituidos principalmente por: un microprocesador, una fuente de energía, una unidad de comunicación y un elemento sensor. Las RSI son usadas en diferentes aplicaciones como son: entornos industriales, domótica, entornos militares, detección ambiental, entre otros [1, 22, 23, 24, 25, 26, 27], teniendo como limitación las características propias de nodo sensor.

Las RSI se forman por un grupo de nodos con capacidad de monitorización y comunicación inalámbrica, formando redes sin necesidad de infraestructura externa, en donde todos los dispositivos se encuentran en igualdad de condiciones. Para ello, este tipo de redes utiliza protocolos de encaminamiento, que se encargan de descubrir las rutas entre los nodos, para lograr que los paquetes de datos lleguen a su destino, aunque éste no sea accesible desde el origen del nodo emisor de los paquetes.

2.1. Arquitectura del sistema

En las RSI, los nodos sensores forman una parte muy importante, éstos tienen una capacidad de cómputo y comunicación limitada y la vida útil de cada nodo sensor depende de el tiempo de vida de la batería que lo alimenta. Como ya se mencionó, los nodos sensores se encuentran dispersos en el área que se desea monitorizar.

La manera en que opera una RSI una vez que los nodos han sido instalados y la red se ha organizado es: primero se realiza la medición analógica sobre el medio, esta información es digitalizada en el nodo, y por último, la información se transmite hacia la estación base. Una vez que la información se encuentra en la estación base, ésta puede ser tratada y almacenada temporalmente, para ser guardada finalmente en un servidor, con una capacidad de almacenamiento mayor, para poder realizar un análisis de los datos.

Una red de sensores inalámbricos, por lo tanto, está compuesta por [1]:

- Nodos inalámbricos
- Puertas de enlace
- Estaciones base

2.1.1. Nodo inalámbrico

Un nodo inalámbrico es un dispositivo electrónico mediante el cual se obtiene información del entorno (temperatura, presión, humedad, entre otros), para que después dicha información sea procesada y transmitida inalámbricamente hacia otro nodo o hacia la estación base. Además los nodos sensores se diseñan de un tamaño reducido, con un consumo muy bajo de energía, un costo bajo y con amplia longitud de emisión.

El hardware de cada uno de estos dispositivos tiene las siguientes partes bien definidas [1]:

Procesador. Se encarga de procesar los datos para poder transmitirlos hacia otro sensor o hacia la estación base. Existen distintos tipos diferentes de productos disponibles en el mercado para ser integrados en un nodo, como microcontroladores, microprocesadores y FPGA (*Field Programmable Gate Array*).

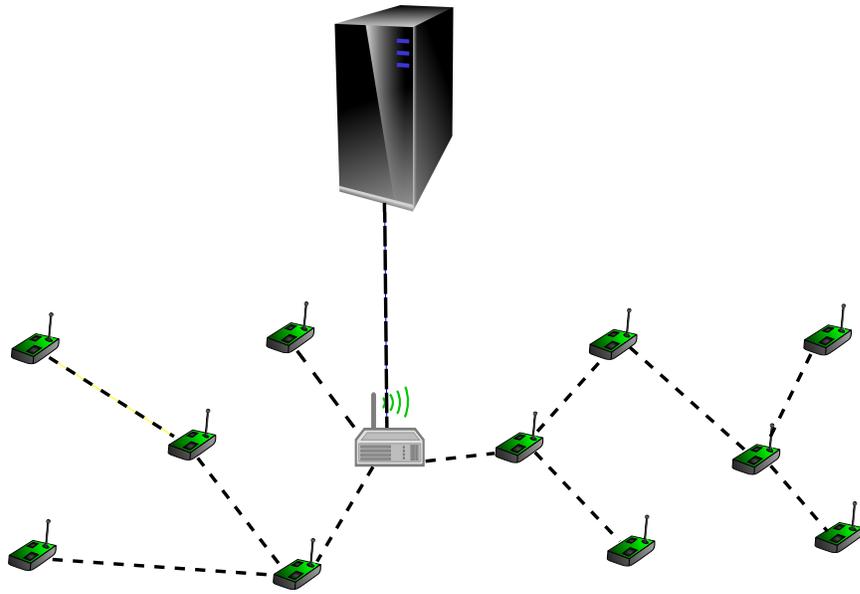


Figura 2.1: Red de sensores inalámbrica.

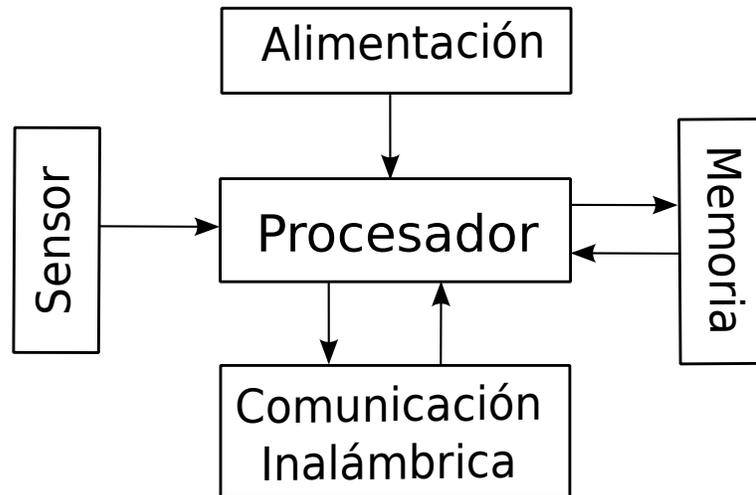


Figura 2.2: Arquitectura de un nodo sensor inalámbrico.

Alimentación. La fuente de alimentación de los nodos sensores es comúnmente baterías que pueden ser recargables o no recargables, o transformadores si se dispone de una toma de corriente. Ya que la vida útil de los sensores depende de la vida que tenga la fuente de alimentación, es necesario contar con una gestión eficiente de la energía. La energía de los nodos sensores es ocupada principalmente por la transmisión de los datos, el procesado de la información y el uso de los sensores, siendo la transmisión de la información la etapa que consume la mayor energía. Actualmente, se estudian sistemas basados en energías renovables para solucionar el problema de la energía en estos nodos. Algunos de estos sistemas están basados en energía solar, termo-generación, energía basada en vibraciones, entre otros.

Comunicación inalámbrica. Se utiliza la comunicación vía radio frecuencia que permite enviar y recibir datos para comunicarse con otros dispositivos dentro de su rango de transmisión. Los nodos utilizan la banda ISM (*Industrial, Scientific and Medical*) que son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en las áreas industrial, científica y médica. Las RSI usan las frecuencias de comunicación que están entre 433 MHz y 2.4 GHz.

Sensores. Los sensores son dispositivos físicos que producen una respuesta medible ante un cambio en un estado físico, como puede ser temperatura o presión. La señal analógica continua detectada, es digitalizada por un convertidor analógico digital y enviada a un procesador que la interpreta y transmite.

Memoria. Desde el punto de gasto de energía, las clases más relevantes de memoria son la memoria integrada en el chip de un microcontrolador y la memoria flash; la memoria RAM (*Random Access Memory*) fuera del chip es raramente usada.

2.1.2. Puerta de enlace

Una puerta de enlace es un dispositivo cuya función es lograr la interconexión entre la red de sensores y una red de datos TCP/IP (*Transmission Control Protocol/Internet Protocol*). Éste dispositivo permite la comunicación de redes con protocolos y arquitecturas diferentes, traduciendo la información de un protocolo en un red inicial al protocolo usado en la red final. El uso de un dispositivo que funciona como puerta de enlace, depende de la aplicación que se este desarrollando, ya que en ocasiones no es necesario tener una conexión entre la red de sensores y una red de datos, ya que la información proveniente de los nodos sensores puede ser almacenada en la estación base para su posterior recuperación y análisis.

2.1.3. Estación base

La estación base es un dispositivo fijo distribuido geográficamente en el área de cobertura de uno o más nodos sensores. Este dispositivo puede ser una computadora común o un sistema empotrado. En una infraestructura utilizada normalmente todos los datos recolectados provenientes de la RSI, son almacenados en una base de datos dentro de un equipo que realiza la función de servidor de datos, desde el cual la información está disponible y puede ser consultada vía remota para poder observar y estudiar los datos.

2.2. Protocolos de encaminamiento de los datos

El estudio de los algoritmos o protocolos de encaminamiento, en las RSI, es de suma importancia ya que de éstos depende la confiabilidad en la entrega de los datos recopilados por la red. Normalmente, los sensores que forman estas redes tienen recursos energéticos limitados, y la capacidad de cómputo que presentan es muy baja. Estas características hacen que los protocolos de encaminamiento desarrollados para redes de sensores deban ser sencillos en términos de computabilidad, y además intenten reducir el consumo de energía [2].

Las RSI son utilizadas en una gran variedad de aplicaciones en las que el medio ambiente y los requisitos técnicos pueden variar considerablemente. Por lo tanto, el diseño de una RSI es por lo general orientado a una aplicación en particular; como consecuencia, las arquitecturas, protocolos y algoritmos varían en cada aplicación. Por tal motivo, es importante realizar clasificaciones de las RSI que nos ayuden a separar los diferentes escenarios que se pueden presentar en el diseño. En la sección 2.2.1 se define una de estas posibles formas de clasificación, la cual está orientada al diseño de este trabajo de investigación.

2.2.1. Clasificación de las redes de sensores inalámbricas

Existen diversas maneras de clasificar las RSI; uno de los criterios para clasificarlas es de acuerdo a la distribución de los nodos; por tanto las RSI pueden ser clasificadas en dos tipos, las estructuradas y no estructuradas [3].

- **RSI no estructuradas.** Es una red de sensores que contiene una colección densa de nodos, que son desplegados de manera aleatoria en un entorno. Una vez desplegada, la

red queda desatendida realizando las funciones para las que se diseñó. En este tipo de redes, tanto el mantenimiento de funciones de red como la conectividad y fallos en la detección, son difíciles debido a la gran cantidad de nodos presentes.

- **RSI estructuradas.** En este tipo de redes todos o algunos de los nodos son colocados en posiciones definidas. La ventaja de este tipo de redes es que los nodos pueden ser desplegados con un menor coste de gestión y mantenimiento de red. Además, se necesitan menos nodos en el despliegue ya que éstos se colocan en posiciones específicas para proporcionar la cobertura deseada.

2.2.2. Diseño de un protocolo de encaminamiento para redes de sensores

El diseño de un protocolo de encaminamiento para redes de sensores está directamente influenciado por la aplicación para la cual se utiliza y por la arquitectura de la red presente, que en el fondo también depende de la aplicación objetivo de la red. Algunos de los puntos más importantes a tener en cuenta en el diseño o elección de un protocolo de encaminamiento para una aplicación determinada son [2]:

- **Despliegue de los nodos.** Puede ser determinista (los nodos se instalan manualmente y se les configuran rutas estáticas) u organizarse por sí mismos (los sensores, después de ser instalados, son capaces de descubrir a los demás dispositivos y construir las rutas al destino).
 - **Movilidad.** Cuando los sensores permanecen estáticos es preferible utilizar protocolos proactivos, en este tipo de protocolos las rutas a todos los nodos están calculadas a priori y se mantienen actualizadas mediante el intercambio periódico de mensajes. Una de las etapas en donde se consume mayor energía es en la etapa de descubrimiento de ruta, y al tener con anticipación las rutas para el encaminamiento de paquetes, se contribuye a conservar la energía en el nodo sensor.
 - **Escala y densidad.** La escala de una red nos da una idea del tamaño de la red (número de nodos presentes), mientras que la densidad nos da una medida de la robustez (tolerancia a fallos).
 - **Modelos de entrega de datos.** Entre los diferentes modelos tenemos el modelo de entrega por tiempo, donde los nodos envían información de manera periódica; el modelo de entrega por ocurrencia de algún suceso o evento; y el modelo de consultas directas a determinados nodos.
-

- **Frecuencia y cantidad de datos.** De la frecuencia y cantidad de datos que envíen los nodos sensores dependerá, en gran medida, el tiempo de vida de la batería.
- **Calidad de servicio QoS.** Puede expresarse en términos de tiempo de vida de la red, fiabilidad de entrega de datos, eficiencia energética, retardo, etc.

2.2.3. Protocolos de encaminamiento

Los protocolos de encaminamiento propuestos para redes de sensores inalámbricas pueden ser divididos en siete categorías [4]:

1. **Protocolos basados en ubicación.** En este tipo de protocolos la dirección de cada nodo sensor se asigna con base en su localización geográfica en la red, con lo cual, se puede conocer la distancia entre dos nodos y el consumo de energía estimado.
 2. **Protocolos centrados en los datos.** En este tipo de protocolos cuando los nodos sensores envían sus datos a la estación base, nodos intermedios pueden agregar información de datos provenientes de múltiples nodos sensores. Este proceso conlleva a un ahorro de energía debido a la disminución de transmisiones necesarias para enviar la información a la estación base.
 3. **Protocolos jerárquicos.** Con estos protocolos la red de sensores se organiza en varios grupos (o clústers) de sensores. Cada uno de estos grupos es dirigido por un nodo especial llamado cabeza de clúster, el cual se encarga de coordinar las actividades de transmisión de datos de todos los sensores de su grupo. Los datos viajan de una capa inferior a una superior permitiendo mover los datos más rápido a la estación base.
 4. **Protocolos basados en la movilidad.** Estos protocolos proponen que, en un escenario con nodos móviles, la estación base se encuentre en movimiento, permitiendo que todos los nodos de la red actúen como puente para los demás nodos. Con esto se logra que todos los nodos tengan un desgaste de energía similar, caso contrario a lo que ocurre con una estación base estática, en donde los nodos mas cercanos a ésta, permanecen en un estado de transmisión por largos periodos de tiempo, ya que además de transmitir los paquetes propios, tienen la función de retransmitir los paquetes de los nodos que se encuentran lejos de la estación base, agotando más rápido su energía.
 5. **Protocolos basados en multiruta.** En el encaminamiento multiruta, cada sensor encuentra los primeros k caminos más cortos hacia la estación base y divide su carga de datos de manera uniforme entre estos caminos.
-

6. **Protocolos basados en heterogeneidad.** Estos protocolos distinguen dos tipos de componentes en la red, los sensores que no tienen alguna restricción de la energía y los sensores que tienen energía limitada; por lo tanto, estos últimos deben utilizar su energía disponible de manera eficiente minimizando su potencia de comunicación de datos y el procesamiento de los mismos.
7. **Protocolos basados en QoS.** Los protocolos basados en QoS (*Quality of Service* - Calidad de Servicio) además de minimizar el consumo de energía, toman en cuenta la calidad de servicio (QoS) en términos de retardo, confiabilidad y tolerancia a fallos. Es decir, estos protocolos buscan el equilibrio entre el consumo de energía y los requisitos de calidad de servicio.

Tener un buen protocolo de encaminamiento no es suficiente para que la red de sensores cumpla con el objetivo de este trabajo de investigación, ya que en un ambiente ruidoso es necesario tener un mecanismo que permita acceder al canal de comunicación de forma ordenada, para minimizar las colisiones y la pérdida de paquetes a causa de la interferencia. En la sección 2.3 se estudia una de las formas en que se clasifican estos mecanismos o protocolos de acceso al medio.

2.3. El control de acceso al medio

La capa MAC (*Medium Access Control* - Control de Acceso al Medio), controla el acceso del nodo al medio de comunicación mediante técnicas como contienda y división de tiempo. Básicamente, debe gestionar los canales de comunicación disponibles para evitar colisiones y errores de transmisión. La mayoría de las soluciones intentan minimizar el consumo de energía al mismo tiempo que proporcionan fiabilidad en la transmisión, algunos ejemplos de estos protocolos son PSM (*Power Save Mode*) de IEEE 802.11, PAMAS (*Power Aware Multi-Access*), Sensor S-MAC, TRAMA (*Traffic Adaptive Medium Access Protocol*), entre otros [21]. En [5], se clasifican los protocolos MAC para redes de sensores inalámbricas en cuatro categorías:

1. **Acceso aleatorio.** En esta clase de protocolos no se restringe el acceso de los nodos al canal de comunicación, esto proporciona una gran flexibilidad para manejar diferentes cantidades de nodos y cargas de tráfico. Además, los nodos no necesitan sincronizar sus relojes, haciendo que estos protocolos sean bastante simples. La desventaja de éstos es que a menudo se desperdicia gran cantidad de energía debido a las colisiones.
 2. **Acceso ranurado.** La idea básica de este protocolo es ahorrar energía, para ello, los nodos están coordinados en un patrón activo/dormido. El tiempo se divide en ranuras
-

de tiempo y los nodos despiertan al comienzo de cada ranura para enviar los mensajes en espera de transmisión. En estos protocolos la posibilidad de colisión es mayor debido a que todas las comunicaciones se agrupan en la pequeña parte activa de la ranura.

3. **Acceso basado en tramas.** Estos protocolos agrupan las ranuras y se programa una agenda para enviar tramas en las ranuras de tiempo, con esto se reducen las colisiones ya que en cada ranura de tiempo sólo hay una par de nodos emisor-receptor activos.
4. **Protocolos híbridos.** Protocolos que son una mezcla de los protocolos de acceso aleatorio y los protocolos basados en tramas. Éstos se adaptan a las fluctuaciones de tráfico y a los cambios de topología.

A pesar de contar con mecanismos o protocolos para controlar el acceso al medio y el encaminamiento de los datos hacia la estación base, los paquetes de datos son susceptibles a corromperse por diferentes causas, como son: ruido en el canal de comunicación, problemas con los circuitos de comunicación, etc. Para tener una red de sensores más robusta y tolerante a fallos, se puede incluir una etapa de control de errores, que nos permita detectar y, en su caso, corregir errores en la trama de datos recibida. En la sección 2.4 podemos ver algunas de las técnicas que se aplican en redes de sensores, como mecanismos de control de errores.

2.4. Mecanismos para el control de errores

Las restricciones de energía en los nodos sensores, hacen que se requieran protocolos de comunicación de bajo consumo energético y que además cumplan con el objetivo de la aplicación para la que fueron desarrollados. Por esta razón, se restringe la energía para efectuar la transmisión de datos, con lo cual se acrecentan los efectos negativos que pueden ocurrir en el canal de comunicación, ya que al ahorrar energía se disminuye la potencia en la etapa de transmisión de datos y se disminuye al mínimo el envío de datos por parte de los nodos. Esto propicia que en un canal de comunicación con ruido existan más pérdidas de paquetes.

En general los mecanismos de control de errores pueden ser clasificados en tres enfoques principales [6]:

- **ARQ (*Automatic Repeat Request* - Demanda de Repetición Automática).** El control de errores ARQ basa su funcionamiento en la retransmisión de los paquetes perdidos para su recuperación. En el control de errores ARQ, se pide la retransmisión de un paquete perdido mediante el envío de mensajes de acuse. Es claro que este

protocolo tiene un costo adicional por las retransmisiones de los paquetes perdidos, lo cual provoca una sobrecarga de la red cuando la calidad del canal de comunicación no es buena. En cambio, si la calidad de canal es buena, entonces el protocolo ARQ resulta ser más eficiente que el protocolo FEC descrito a continuación. En aplicaciones de redes de sensores inalámbricas el protocolo ARQ tiene un uso limitado, debido a la prioridad de conservación de la energía y a la poca capacidad de procesamiento que tienen los nodos.

- **FEC (*Forward Error Correction* - Corrección de Error hacia Adelante).** La corrección de errores FEC añade redundancia a los paquetes a transmitir, por lo que los paquetes pueden ser recibidos por los nodos destino y la información puede recuperarse aunque ésta llegue con errores, siempre y cuando el número de bits con errores sea limitado. Es claro que la adición de redundancia en los paquetes, ocasiona una carga al estado de la red. En una RSI se debe de tomar en cuenta que el proceso de codificación y decodificación consume energía en los nodos. Existen códigos FEC que se pueden optimizar para un tamaño de paquetes específico, condiciones de canal y requisitos de fiabilidad, tales como Red Solomon, códigos BCH y códigos convolucionales.
- **HARQ (*Hybrid ARQ*).** Este tipo de protocolo pretende explotar las ventajas de ARQ y FEC, implementando una elasticidad en el error con retransmisiones. Cuando un paquete es recibido con error, el receptor envía un mensaje NACK (Negative Acknowledgement) al transmisor, el cual envía de nuevo el paquete con un código FEC más robusto.

2.5. Redes de sensores inalámbricas y el ruido

En ambientes con ruido, el desempeño de una red de sensores inalámbrica se ve mermado en diferentes aspectos como son: la latencia de una comunicación de extremo a extremo se incrementa, el uso del procesador aumenta, el número de paquetes perdidos y de retransmisiones aumenta, y esto conlleva a un consumo excesivo de energía, entre otros [17].

En [17] se hace un estudio del ruido en ambientes industriales, el cual está basado en el uso de un analizador de espectros de rango 150 kHz a 7.1 GHz y dos antenas de gran ancho de banda que cubren rangos de 30-3000 MHz. Se encontró que la intensidad de señal se puede ver interrumpida debido a alteraciones como la reflexión de onda en las paredes y pisos, además de las interferencias de las bandas ISM y del ruido generado por los equipos pesados o maquinaria, causando interferencias destructivas. De esta manera, es necesario contar con mecanismos que conserven la integridad de los datos.

En [18] se presentan algunas de las fuentes de interferencia más comunes en ambientes industriales. Éstas las podemos observar en la tabla 2.1.

Interferencias de gran ancho de banda	Interferencias de ancho de banda angosto
Motores	Transmisores de radio y TV
Inversores, circuitos SCR	Zumbido de líneas eléctricas
Interruptores eléctricos	Generadores de señales
Computadoras	Osciladores locales
Sistemas de ignición	Sistemas UPS
Reguladores de voltaje	Equipos de medición
Pulsos electromagnéticos (Rayos)	Equipos de microondas y de ultrasonido
Lámparas de arco o vapor	Balastro electrónico
Generador de pulsos	Equipo médico
Termostatos	Sistemas con microprocesadores
Aparatos de soldadura	Transmisores celulares y localizadores
Convertidores de frecuencia	Generadores de alta frecuencia

Tabla 2.1: Fuentes de interferencia (basado en [18].)

Protocolo propuesto

En algunas ocasiones se puede pensar que para ciertas aplicaciones es indiferente el uso de RSI estructuradas o no estructuradas, sin embargo, debemos de tomar en cuenta que por ejemplo, en una aplicación de daños estructurales, si se instala una RSI no estructurada en donde los nodos sensores se ubican en posiciones arbitrarias, no se tendría una fiabilidad en los datos recabados, ya que la información que se necesita obtener se debe de tomar en puntos específicos en la estructura del edificio.

Por el contrario, en aplicaciones en donde se utilizan RSI no estructuradas, el desempeño se ve mejorado, al ser un sistema más fiable en recolección de datos. Sin embargo, esto en muchas ocasiones es difícil de realizar, ya que las aplicaciones en donde se utilizan RSI no estructuradas son aquellas en las que las zonas de cobertura son de difícil acceso, lo cual impide planear una posición exacta de los sensores.

En el desarrollo de este proyecto, se adopta el modelo de una RSI estructurada, debido a que nuestro caso de estudio requiere una topología estática de sensores.

A continuación, se definen algunos puntos importantes que sirven como base para la elaboración de este trabajo de investigación.

- Los nodos sensores en la red permanecen en una posición definida, es decir, se trata de una red de sensores estructurada. Esto permite tener un mejor control de la red, debido a que los nodos sensores son colocados en el lugar óptimo para lograr la cobertura deseada. La estación base también permanece estática.
- La estación base es la que se encarga de recibir los datos de todos los nodos sensores de la red.

- La red de sensores trabaja en ambientes con un nivel de ruido alto, lo que provoca una constante pérdida de paquetes en toda la red.
- La propuesta esta orientada a la monitorización de señales físicas que varían muy poco en el tiempo, tal como la temperatura ambiente, humedad, presión atmosférica, entre otros.

En este trabajo de investigación se propone un protocolo, nombrado EMT (Encaminamiento por Multiplexión de Tiempo), con el cual una red de sensores es capaz de operar en ambientes con nivel de alto de ruido, el cual provoca la pérdida de los paquetes o la corrupción de los datos. Para ello, se propone un protocolo de acceso al medio, un protocolo de encaminamiento de datos y un mecanismo de detección de errores. En este caso de estudio, se tiene como prioridad que todos los nodos de la red entreguen los datos recolectados en un tiempo máximo, es decir, la fiabilidad en la entrega de paquetes que se busca obtener, no está basada en tener la mayor cantidad de paquetes en un conteo global de la red, sino que todos los nodos entreguen por lo menos un paquete, en un tiempo máximo.

3.1. Topología de la red de sensores

Una parte importante de este trabajo como hemos mencionado, es la topología de la red, por lo cual primero definimos la topología utilizada en este estudio con la cual evaluamos el protocolo propuesto.

- La red de sensores está compuesta por cien nodos sensores estáticos y una estación base que también se encuentra estática.
 - Los nodos se encuentran divididos en dos líneas paralelas separadas por diez metros.
 - La distancia entre nodos consecutivos de la misma línea es de diez metros, por lo tanto el escenario de simulación tiene una longitud de quinientos metros.
 - La estación base está en medio de la red de sensores.
-

En la figura 3.1, se muestra el escenario propuesto.

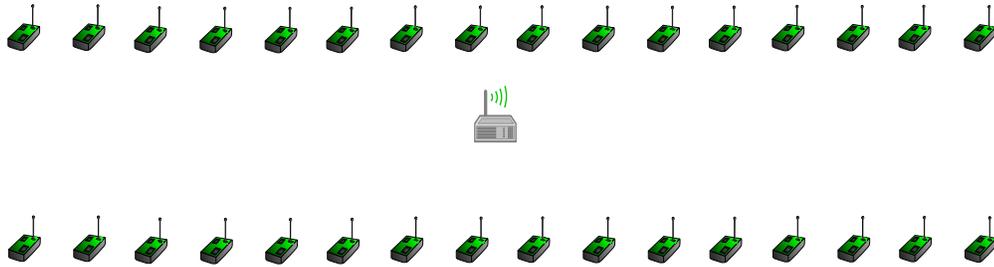


Figura 3.1: Topología - red de sensores inalámbrica.

3.2. Protocolo de acceso al medio

El control de acceso al medio es otra parte muy importante en este trabajo, debido a que se busca una forma ordenada de acceso al medio, en la que los nodos sensores puedan enviar la información registrada, además de evitar al máximo las pérdidas provocadas por colisiones.

En un escenario con muchos nodos, como el planteado en este trabajo, un protocolo de acceso aleatorio ocasiona múltiples colisiones, por tal motivo los protocolos de este tipo están descartados en este trabajo de investigación. Por otro lado, los protocolos de acceso ranurado o basados en tramas, parecen ser una mejor opción.

Nuestro protocolo está basado en TDMA, el cual es un procedimiento que divide el canal de comunicación en varias ranuras de tiempo; y a cada nodo sensor se le asigna una ranura de tiempo específica para la transmisión, lo que hace posible que varios nodos sensores utilicen un mismo canal de comunicación simultáneamente sin interferir entre sí. Además este protocolo reduce el uso de memoria, CPU y energía, siendo sencillo de implementar.

Para asignar el tiempo en el que los nodos transmiten sus datos, nos basamos en un identificador lógico de los nodos que es único en toda la red de sensores. En la figura 3.2 se puede observar cómo los nodos que componen la red utilizan una ranura de tiempo definida para poder transmitir su información hacia la estación base evitando las colisiones ocasionadas por la transmisión de mensajes simultáneos por dos o más nodos.

Cabe mencionar también que los nodos pueden retransmitir la información proveniente de otros nodos a la estación base.

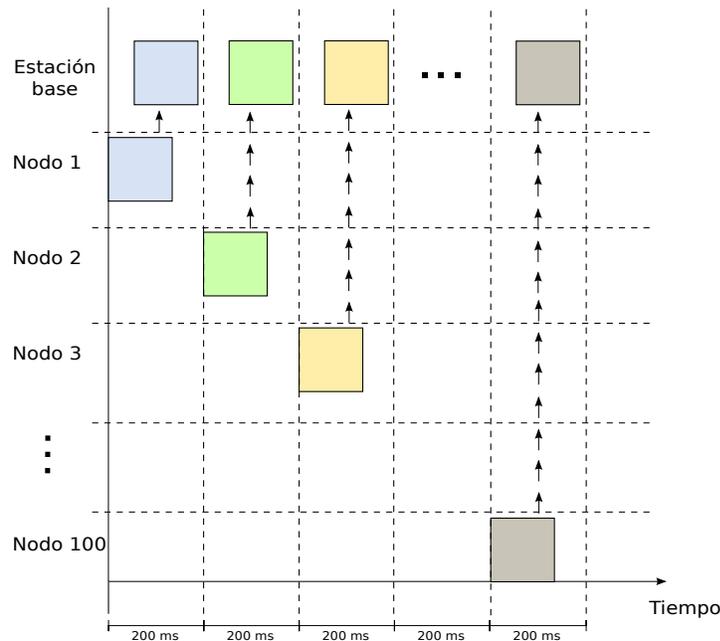


Figura 3.2: Control de acceso al medio.

3.3. Protocolo de encaminamiento de datos

El protocolo de encaminamiento propuesto en este trabajo de investigación, es un protocolo que busca garantizar una mínima entrega de datos de todos los nodos de la red y un retraso de extremo a extremo máximo. En la literatura podemos encontrar protocolos para el encaminamiento de datos en redes de sensores tales como como: AODV (*Ad Hoc On-Demand Distance Vector*) [7] [8], LEACH (*Low-Energy Adaptive Clustering Hierarchy*) [20], entre otros. Sin embargo estos protocolos fueron pensados para aplicaciones o escenarios en donde la fiabilidad de entrega de datos no se ve comprometida por el ruido en el ambiente.

Sin embargo, para este caso de estudio es necesario un protocolo que presente un desempeño mínimo adecuado a pesar de la presencia de un alto porcentaje de pérdidas de paquetes. Para ello, se propone un protocolo que no depende de mensajes de acuse para su operación ya que, en un ambiente ruidoso, estos mensajes también son propensos a perderse o corromperse, haciendo que los protocolos convencionales se desestabilicen y de esta manera disminuya su desempeño.

En este trabajo de investigación, se plantea una red de sensores para una aplicación en tiempo real, es decir, que es importante para la aplicación recibir los datos generados por los sensores con un retraso no mayor al umbral. Además, dado que la aplicación tolera la

pérdida de datos y los datos llegan continuamente de los sensores, no es necesario recuperar las pérdidas, ya que la pérdida de un paquete es cubierta con la recepción del siguiente paquete. Por tal motivo, se eliminan los mensajes de acuse de paquete recibido y los mensajes de petición de retransmisión de paquetes; quedando sólo los mensajes de transmisión de datos y los de descubrimiento de rutas. Lo que provoca un consumo bajo de energía, memoria y CPU. Además resulta en un protocolo simple de implementar.

Por otra parte, cuando la red se inicia, se propone tener un tiempo de configuración de red, el cual permite que los nodos armen su tabla de rutas gracias al siguiente procedimiento:

1. La estación base difunde un primer mensaje, indicando que se encuentra lista para recibir información de los nodos sensores, a toda la red.
2. Los nodos cercanos a la estación base que son capaces de recibir este mensaje, comienzan a construir su tabla de rutas con comunicación directa a la estación base. En ese momento, estos nodos mandan un primer mensaje, que inicia el proceso de descubrimiento de ruta en los nodos que no ven la estación base. Para éstos utilizaremos el término "nodo posterior".
3. Los nodos posteriores reciben el mensaje de descubrimiento de ruta y colocan al nodo fuente del mensaje como su primer elemento en la tabla de rutas.
4. El proceso se repite y de esta forma los nodos sensores obtienen en su tabla de rutas empleando como encaminador a los nodos que están más cerca de la estación base.

Debido a que es una red estructurada, tenemos la certeza que al construir una primera tabla de rutas ésta debe de mantenerse ya que los nodos no cambian de posición, por lo tanto, no es necesario tener un mecanismo para actualizar la tabla de rutas de los nodos sensores.

En un ambiente ruidoso, los paquetes son más susceptibles a perderse o a corromperse. Además, si se usan protocolos que intercambian mensajes periódicamente para mantener actualizadas las tablas de rutas o para informar de la recepción de mensajes, estos mensajes provocan un efecto que incrementa la interferencia en las comunicaciones y por ende, resulta en más pérdidas de paquetes. Además, si se pierden los mensajes de acuse se provoca un mal funcionamiento de los protocolos que dependen de ellos. En la figura 3.3, se puede observar cómo un mensaje enviado por algún nodo, puede causar interferencia con otros nodos que se encuentran cerca de su rango de transmisión. Este tipo de interferencia es de mayor importancia en un ambiente que se considera ruidoso, debido a que afecta aún más las condiciones de comunicación entre los nodos y la estación base.

Una vez que el protocolo de encaminamiento de datos se ha iniciado y se construyeron las tablas de rutas para los nodos, los mensajes serán transmitidos por los nodos sensores que

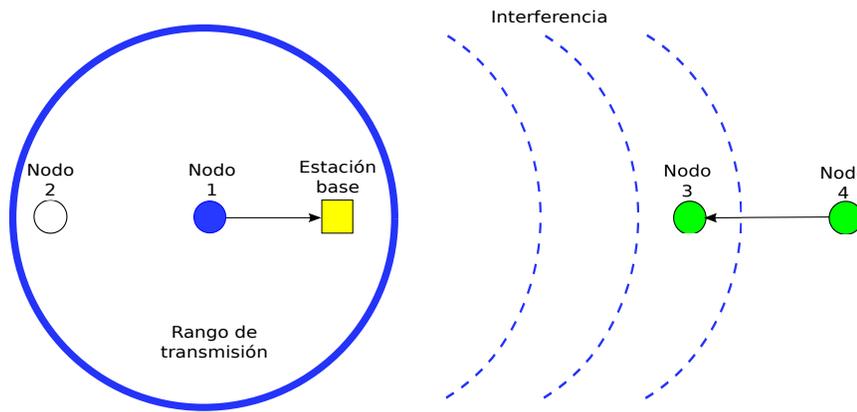


Figura 3.3: Inteferencia casuada por los nodos de la red.

componen la red, en la ranura de tiempo asignada de acuerdo a su identificador dentro de la red. En este trabajo de investigación se contemplan escenarios en los que, a causa del ruido, la tabla de rutas no se construye de acuerdo a la cercanía de los nodos, sino con base en el orden en que los nodos reciben mensajes de sus vecinos, y una vez que un nodo recibe un mensaje para retransmitirlo hacia la estación base, éste lo hará inmediatamente, es decir, no espera a que llegue su ranura de tiempo para poder transmitirlo. Sino que utiliza las rebanadas libres que ha ido descubriendo a lo largo de la operación de la red. En la figura 3.4, podemos ver un ejemplo de cómo el protocolo de encaminamiento y el de acceso al medio trabajan de forma simultánea para lograr la comunicación de la red de sensores con la estación base.

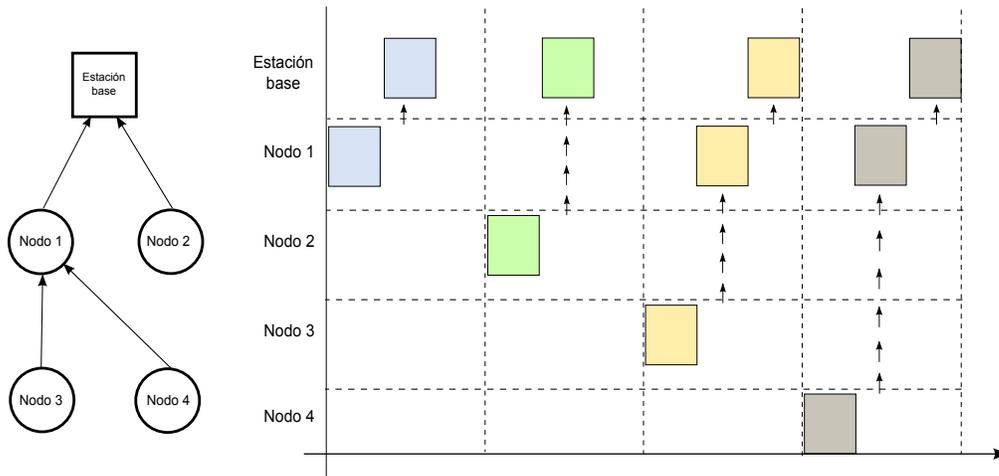


Figura 3.4: Encaminamiento de los datos.

Evaluación de desempeño

En este capítulo se muestra la forma en que se realizó la evaluación de desempeño del protocolo de comunicación propuesto. Para ello fue necesario dividir la evaluación en dos partes. En la primera se realiza la simulación de una red de sensores más compleja en número de nodos y dimensiones, que integra los mecanismos de acceso al medio, así como los de encaminamiento de paquetes. En la segunda parte se realiza la implementación de los protocolos de acceso al medio y de detección de errores en un circuito prototipo.

4.1. Simulación de una red inalámbrica de sensores

4.1.1. El simulador de redes NS-2

El simulador NS-2 (*Network Simulator - 2*) [9] es un simulador de redes basado en eventos discretos. El simulador NS-2 representa una alternativa importante para la investigación en redes y comunicaciones. El simulador NS-2 tiene soporte para redes cableadas e inalámbricas, además, permite ajustar distintos parámetros en la simulación, con lo que se pueden generar escenarios robustos que permiten el estudio de diferentes protocolos de comunicación. NS-2 permite la simulación de diferentes tipos de protocolos de encaminamiento, generación de tráfico como TCP y UDP, protocolos de capa de aplicación como FTP, entre muchas otras características.

El simulador está implementado en lenguaje OTcl y C++; OTcl sirve como intérprete de órdenes que el usuario programa al momento de crear los escenarios de simulación, la

implementación de todas las clases del simulador está escrita en el lenguaje C++, por lo tanto, cuando se necesitan realizar modificaciones e implementaciones de nuevos protocolos, se tiene que modificar el código fuente escrito en C++ y anexar el nuevo módulo a los existentes para agregar nuevas funcionalidades al simulador. Esta ventaja se debe a que NS-2 está bajo licencia GPL (*General Public License*), la cual nos permite realizar modificaciones al código fuente proporcionado por los desarrolladores del simulador.

La interfaz hacia el usuario de NS-2 está basada en el lenguaje TCL (*Tool Command Language* - Lenguaje de herramientas de comando), que es un lenguaje que se usa principalmente en aplicaciones script, e interfaces gráficas. Sus reglas sintácticas son extremadamente simples y posee reglas de alcance dinámico. Permite escribir código fácil de mantener, ya que los scripts TCL son a menudo más compactos y legibles que los programas funcionalmente equivalentes en otros lenguajes de programación. Por medio de scripts TCL creamos los escenarios y definimos los parámetros de las simulaciones realizadas en NS-2 durante el desarrollo de este trabajo. Un manual completo de la programación de script TCL en NS-2 se encuentra en [16]. Una vez creado el script y después de ejecutarlo se genera un archivo de texto denominado archivo de traza, el cual contiene toda la información de la ejecución de la simulación, como son: tiempos de envío y recepción de paquetes entre los nodos, paquetes perdidos, mensajes de acuse de los nodos, mensajes de configuración, entre muchos otros. El análisis de los archivos de traza, es lo que permite obtener los resultados de la simulación.

4.1.2. Simulación del protocolo de encaminamiento y acceso al medio

Para evaluar los protocolos de acceso al medio y de encaminamiento de paquetes se realizó la simulación de los mismos mediante el simulador para redes NS-2, en su versión 2.34, para ello se compiló el simulador NS-2 con nuestro protocolo de encaminamiento, adicionalmente se programó el acceso al medio mediante el script de simulación TCL.

4.1.3. Escenario de simulación

El escenario de simulación propuesto, consta de cien nodos y una estación base. Los nodos se ubican en dos líneas paralelas separadas diez metros, cada línea con cincuenta nodos separados entre sí por diez metros; la estación base se encuentra justo a la mitad del escenario de simulación. El escenario de simulación por tanto tiene una longitud de quinientos metros y una anchura de diez metros, pero nada impide que la red se expanda a la redonda

cubriendo una área circular. El escenario propuesto para las simulaciones se puede observar en la figura 4.1.

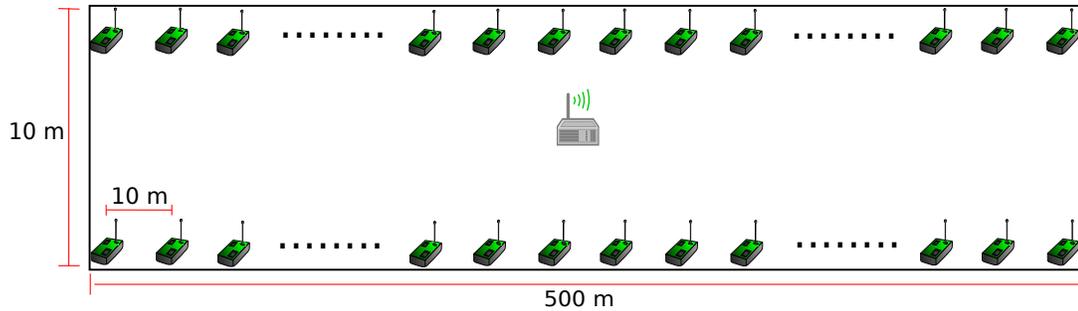


Figura 4.1: Escenario de simulación.

Para crear los escenarios se programó un script que asigna a cada posición del escenario un identificador lógico único y aleatorio en el rango de 1 a 100, cabe mencionar, que para poder asegurar que estos escenarios son completamente aleatorios primero se generaron cien números aleatorios que sirven como semilla del script.

4.1.4. Parámetros de simulación

Para realizar la simulación con el escenario planteado y una vez que el protocolo fue compilado en NS-2, fue necesario definir los parámetros que se utilizan en dicha simulación. Los parámetros principales en la simulación se muestran en la tabla 4.1.

Parámetro	Valor
Modelo de propagación	TwoRayGround
Tamaño de la cola de espera	10
Número de nodos	101
Tipo de tráfico	CBR
Tiempo de simulación	503 s
Tamaño de los paquetes	6 bytes
Paquetes por ranura de tiempo	1
Duración de la ranura de tiempo	200 ms

Tabla 4.1: Parámetros de simulación.

En cada simulación se considera un tiempo total de 503 segundos, de los cuales los primeros 2 segundos son utilizados como una fase de inicialización de la red en general, y el tiempo

restante se consume en la transmisión de mensajes de los nodos. Este tiempo fue calculado con base en el tiempo en que todos los nodos de la red tardan en mandar 25 mensajes de datos hacia la estación base; recordemos que en nuestro esquema las ranuras de tiempo son de 200 ms y durante este tiempo sólo un nodo puede transmitir un mensaje de datos propio.

El mensaje de datos de los nodos es de tipo UDP (*User Datagram Protocol*) con una longitud total de 14 Bytes, 8 Bytes para los campos de cabecera y 6 Bytes de datos, como se muestra en la figura 4.2, cada nodo transmite sólo un mensaje de datos en la ranura de tiempo dedicada a él.

byte	1 - 2	3 - 4
1	Puerto origen	Puerto destino
5	Longitud de mensaje	Suma de verificación
9	Datos	Datos
13	Datos	

Figura 4.2: Formato del mensaje de datos UDP.

Una parte importante de este trabajo, es la presencia de ruido en los escenarios de simulación, para ello, en los párrafos siguientes presentamos la forma en la cual se simuló el ruido en nuestro caso de estudio.

En este trabajo de investigación, no modelamos las causas del ruido en la red de sensores, solo recreamos el comportamiento a nivel paquete causado por este ruido. Nuestro caso de estudio, como hemos mencionado es para ambientes ruidosos, en los cuales el ruido no se presenta en forma de ráfagas, sino en forma continua, y por lo tanto podemos utilizar el método de *muestreo simple* [10]. Este enfoque tiene la ventaja que, tanto su implementación, como la toma de muestras, se puede conseguir de una manera rápida.

Suponiendo que cada muestra de ruido no está correlacionada, la simulación del ruido se reduce a la generación de variables aleatorias, para ello se incluye en el código de recepción de paquete de nuestro protocolo una rutina que calcula un número aleatorio entre 1 y 100, y éste es comparado con un porcentaje de pérdidas definido, con lo cual se decide si el paquete se considera como perdido o no.

La implementación de esta rutina se realizó directamente en el protocolo que se programó en NS-2, específicamente en el procedimiento que se encarga de tratar los paquetes recibidos. En el Algoritmo 1, se puede observar cómo implementamos las pérdidas de los paquetes.

Algoritmo 1 Algoritmo utilizado para inducir pérdidas de paquetes

```
1: porcentaje de pérdidas = 50 {Número entre 0 y 100}
2: Proc: recepción de paquetes(paquete)
3: probabilidad de pérdida = Random::integer100
4: si (probabilidad de pérdida < porcentaje de pérdidas) entonces
5:     pierde paquete
6:     devolver
7: si no
8:     procesa paquete
9: fin si
```

4.1.5. Script de simulación tcl

Una vez que el protocolo fue implementado y el simulador de redes NS-2 fue recompilado, se realizó el script de simulación en lenguaje TCL, en este script se definen los parámetros de simulación vistos en la sección 4.1.4, además, se implementa el algoritmo que simula el protocolo de acceso al medio que se desarrolló en este trabajo y que describimos en la sección 3.2. El script de simulación completo se puede encontrar en el apéndice A.

4.2. Implementación de la red de nodos sensores

Para poder evaluar el mecanismo de detección de errores y el control de acceso al medio, se implementaron físicamente algunos nodos sensores, es decir, se diseñaron y construyeron nodos sensores. Para ello, se utilizó como procesador el PIC18F26J50 de Microchip [12] y como módulos de comunicación, los circuitos TXM-433-LR-S [13] y RXM-433-LR-S [14] del fabricante LINX, con lo cual se tiene un circuito sencillo pero funcional como sensor de prueba. Cabe mencionar que en este caso no hay un transductor captando alguna señal física, sino una rutina *software* que genera datos que emulan la operación de un transductor. En el apéndice C se muestra el diagrama esquemático completo del circuito sensor desarrollado. En cuanto al programa que corre en los nodos físicos se utilizó el compilador MikroC v6.0.

Para lograr una mejor sincronización de los nodos en la red de sensores, se implementó una etapa de codificación Manchester [11], con esto logramos tener una etapa de detección de errores en la recepción de mensajes.

En la codificación Manchester, cada periodo de un bit se divide en dos intervalos iguales. Un bit binario de valor 1 se transmite con valor de tensión alto en el primer intervalo y un

valor bajo en el segundo. Un bit 0 se envía al contrario, es decir, una tensión baja seguida de un nivel de tensión alto. Este esquema asegura que todos los bits presentan una transición en la parte media, proporcionando así una excelente sincronización entre el receptor y el transmisor. Una desventaja de este tipo de transmisión, es que se necesita el doble del ancho de banda para la misma información que el método convencional. En la figura 4.3, podemos ver un ejemplo del funcionamiento de la codificación Manchester.

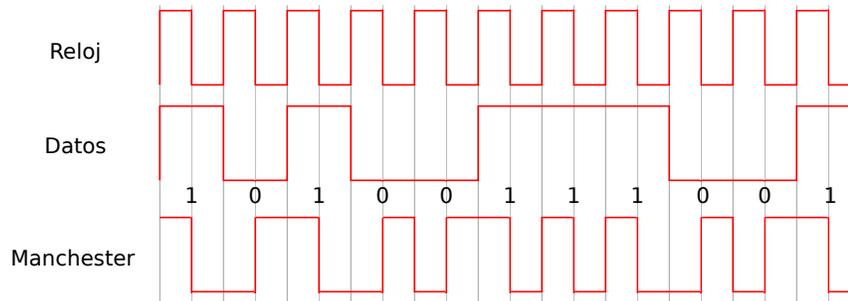


Figura 4.3: Ejemplo de codificación Manchester.

Para los nodos físicos la codificación Manchester se realizó mediante la implementación una tabla Hash, por medio de un arreglo en lenguaje C, usando al índice del arreglo como clave de la tabla, y el valor del arreglo como dato de la tabla. Como el mensaje se envía por bytes, y al codificar en Manchester se duplica el número de bits, de un byte se obtienen dos bytes, por lo que la tabla Hash se diseñó sólo para codificar un byte a la vez, es decir, para la transmisión de un mensaje son necesarias dos codificaciones una por cada nibble (cada byte a transmitir se divide en nibbles y se codifica cada nibble obteniéndose así un byte por cada nibble). En la tabla 4.2 se muestra la tabla usada para la codificación de los nibbles.

Dado que el protocolo debe permitir a los sistemas el intercambio de mensajes entre ellos y por tanto el reconocimiento del origen y el destino de cada transmisión, se incluye en la trama un byte con el cual se indica el nodo fuente del mensaje y un byte que indica el nodo destino; el campo de mensaje está formado por dos bytes y por último, para poder verificar la integridad de los mensajes y poder desechar mensajes alterados por el ruido, se incluirá un byte para la suma de verificación, el cual fue implementado mediante software en la programación del PIC18F26J50. Ya que la aplicación que se estudia es para una red sensores en tiempo real, se hace uso de mensajes parecidos a UDP, ya que para su uso no es necesario tener una conexión previa entre los nodos, además, en la cabecera de este mensaje se integra suficiente información de direccionamiento, quedando una trama formada por cinco bytes que codificados se convierten en diez, como se muestra en la figura 4.4.

Como se describe en la sección 3.2, el protocolo de acceso al medio que se implementa en este trabajo de investigación está basado en ranuras de tiempo. Las ranuras de tiempo definidas son de 200 ms, en las cuales cada nodo envía el mensaje que tiene almacenado. Para

Nibble	Byte codificado en Manchester	Byte en hexadecimal
0000	01010101	55
0001	01010110	56
0010	01011001	59
0011	01011010	5A
0100	01100101	65
0101	01100110	66
0110	01101001	69
0111	01101010	6A
1000	10010101	95
1001	10010110	96
1010	10011001	99
1011	10011010	9A
1100	10100101	A5
1101	10100110	A6
1110	10101001	A9
1111	10101010	AA

Tabla 4.2: Valores utilizados para la codificación de nibbles

byte	1	2
1	Nodo origen	Nodo destino
3	Datos	Datos
5	Suma de verificación	

Figura 4.4: Trama del mensaje en los circuitos de prueba.

que cada nodo sensor transmita el mensaje de datos en su ranura de tiempo, es necesario tener una etapa de sincronización, la cual se inicia en la estación base y consiste en el envío de una trama de sincronización que porta el $ID = 0$. Ésta es recibida por los nodos sensores más cercanos y reconocido como el inicio de una trama de sincronía; los nodos sensores que no son capaces de recibir el mensaje de sincronía de la estación base, toman cualquier mensaje que puedan recibir como mensaje de sincronía. Cuando un nodo recibe un mensaje de sincronía, ya sea de la estación base o de algún nodo cercano, éste usa la información del ID fuente del mensaje, para crear su tabla de rutas, poniendo como primer elemento de la tabla a la fuente del mensaje de sincronía. De esta forma queda establecido el circuito de comunicación en la red de nodos sensores.

Bajo este esquema, un nodo está restringido a transmitir sólo en la ranura de tiempo que corresponde. Tomando en cuenta que en el protocolo propuesto cada ranura de tiempo dura 200 ms, un nodo debe de esperar $ID * 200ms$ antes de tener derecho a transmitir: el instante de inicio de un periodo de transmisión es marcado por la recepción de una trama con $ID = 0$, o trama de sincronización.

La ecuación general que define al tiempo que un nodo deberá esperar antes de transmitir una vez recibido el mensaje de sincronía es:

$$t_{de\ espera} = (id_{receptor} - id_0)(200ms)$$

De forma general podemos definir la programación de la estación base con el autómata mostrado en la Figura 4.5.

La operación de la estación base inicia cuando envía un mensaje de sincronía a toda la red. En este momento se inicia un temporizador, cuando éste expira se produce una interrupción, la cual envía un nuevo mensaje de sincronía a la red. Esto se hace para tener la red sincronizada en todo momento.

Al iniciarse la red, la estación base entra en una rutina de recepción de mensajes provenientes de los nodos de la red, cuando recibe cada mensaje lo procesa y queda en espera de un nuevo mensaje.

La programación en los nodos sensores tiene variantes a la realizada en la estación base, ya que éstos cumplen con funciones distintas. En la figura 4.6 se muestra el autómata que sigue esta programación.

La operación de los nodos, se inicia cuando estos reciben el primer mensaje de sincronía, esto ocasiona que se haga un ajuste del temporizador de transmisión, posteriormente se encola

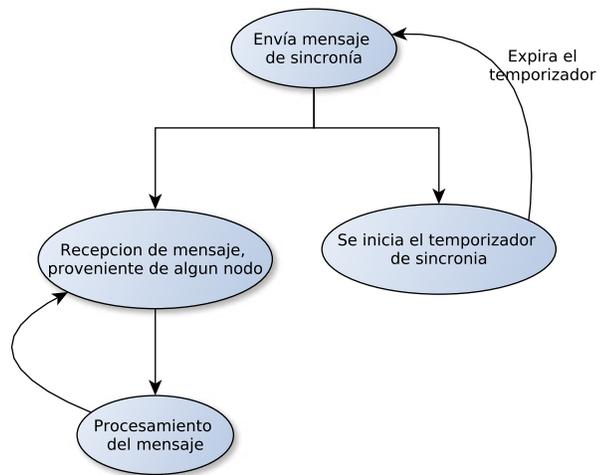


Figura 4.5: Autómata de operación de la estación base.

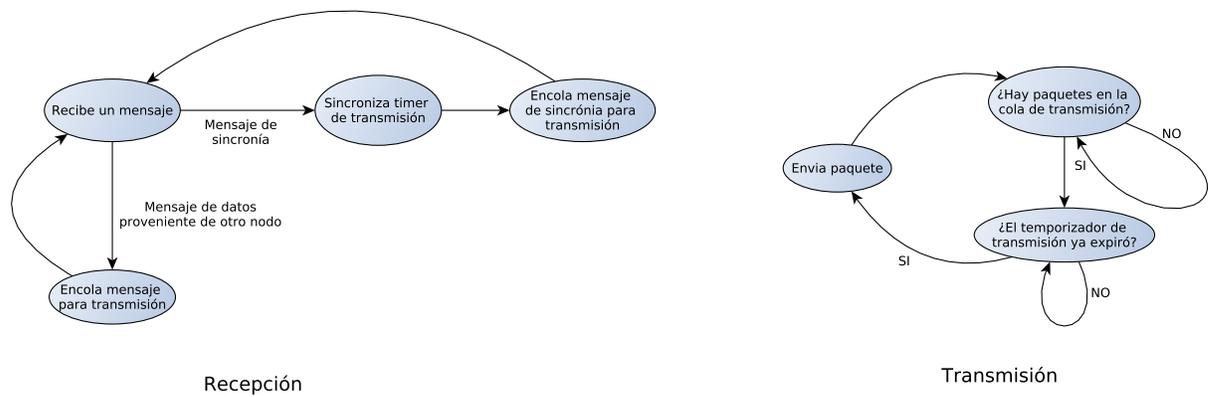


Figura 4.6: Autómata de la operación de los nodos.

el mensaje de sincronía para ser transmitido a los demás nodos. El reajuste del temporizador de transmisión se realiza periódicamente al recibir un mensaje de sincronía, de la estación base o de algún nodo cercano.

Cuando el temporizador de transmisión expira, se puede transmitir un mensaje de la cola de transmisión, ya que en ese momento el nodo se encuentra en la rebanada de tiempo asignada a él.

En términos generales, el protocolo de comunicación consiste en la retransmisión de mensajes por parte de los nodos sensores con la finalidad de hacer llegar los mensajes de todos los nodos de la red hasta la estación base, la cual dentro de la implementación, con la intención de observar el comportamiento del sistema, se conectó mediante un UART (*Universal Asynchronous Receiver-Transmitter* - Transmisor-Receptor Asíncrono Universal) a una PC, la cual muestra los mensajes recibidos por la base a través del software *hyperterminal* de Windows. En la figura 4.7 se muestra el diagrama general de la red de sensores que se implementó para las pruebas.

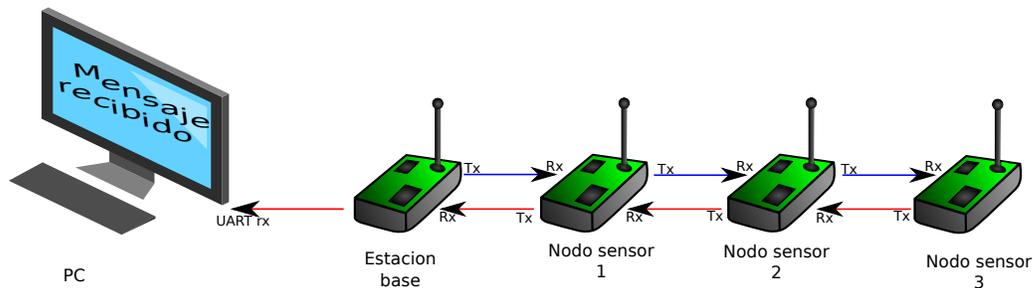


Figura 4.7: Red de sensores de prueba.

Resultados obtenidos en la simulación y en la red prototipo

5.1. Resultados de la simulación

En este capítulo se muestran los resultados obtenidos después de la simulación del protocolo en NS-2. Para tener un punto de comparación de nuestro protocolo, se realizó la simulación en cuatro escenarios diferentes. La topología del escenario fue la misma en todos los casos así como los parámetros definidos en la sección 4.1.4, sin embargo se simularon dos protocolos diferentes de encaminamiento, uno de ellos es AODV, con lo cual se da un acercamiento al funcionamiento de la tecnología Zigbee, la cual es utilizada en dispositivos sensores como: *EasyBee*, *ETRX1*, *TynnyOS*, entre otros. El otro protocolo de encaminamiento simulado es el propuesto en esta tesis denominado como protocolo EMT (Encaminamiento por Multiplexión de Tiempo).

Para cada uno de estos escenarios se realizaron además dos simulaciones variando el rango de transmisión de los nodos. Se generaron resultados tanto para un rango de transmisión de quince metros como de cuarenta metros. Por tanto tenemos cuatro escenarios base para la simulación como se muestra en la tabla 5.1.

Ahora con estos cuatro escenarios definidos se realizó la simulación variando el porcentaje de pérdidas de los paquetes en cada uno de los casos. La variación fue en un rango de 0 a 50, con un incremento de 5, es decir, para cada escenario propuesto se realizaron once simulaciones con diferente valor de porcentaje de pérdidas.

Escenario	Protocolo de encaminamiento	Rango de transmisión
1	AODV	15 m
2	AODV	40 m
3	EMT	15 m
4	EMT	40 m

Tabla 5.1: Escenarios de simulación

Cada una de estas simulaciones fue ejecutada cien veces variando el valor del identificador de cada nodo en la red, usando el método descrito en la sección 4.1.3. Una vez que el simulador de redes NS-2 creó el archivo de trazas para cada una de las 100 simulaciones de cada escenario, es necesario extraer los datos relevantes de estos archivos, esto con el fin de observar de una manera más sencilla los resultados de la simulación. La forma de extraer estos resultados de los archivos de traza de NS-2 se realizó mediante un script C Shell, el cual permite la ejecución de comandos *awk*, los cuales nos permiten hacer un filtrado de los datos contenidos en los archivos de traza. En el apéndice B se muestran los script usados para dicho filtrado.

5.2. Porcentaje de pérdidas

Como primer elemento de comparación se usó el porcentaje de pérdidas de paquetes en la simulación, para ello se divide el escenario en regiones tomando como centro la estación base, y dividimos el escenario en 25 regiones de acuerdo a la distancia que tienen los nodos de la estación base. En la figura 5.1 se muestran las regiones en las que se dividió el escenario de simulación.

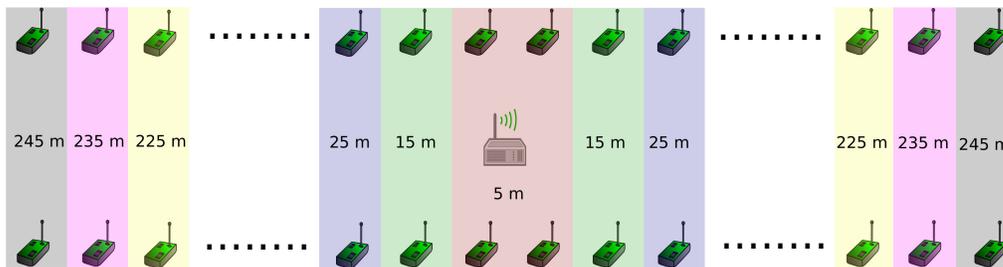


Figura 5.1: Distancias de los nodos con respecto a la estación base

Como podemos observar en la figura 5.2, al usar el protocolo de encaminamiento AODV, las pérdidas de los paquetes se mantienen constantes para una distancia de transmisión hasta

de 245 metros entre la estación base y los nodos. Esto sin importar el porcentaje de pérdidas programado en el protocolo. En el caso del protocolo de encaminamiento desarrollado en esta tesis podemos observar (figura 5.3) que las pérdidas aumentan de manera significativa con respecto a la distancia entre los nodos y la estación base y del porcentaje de pérdidas. es importante mencionar que el porcentaje de pérdidas se va variando, hasta alcanzar casi el 100 % de pérdidas totales.

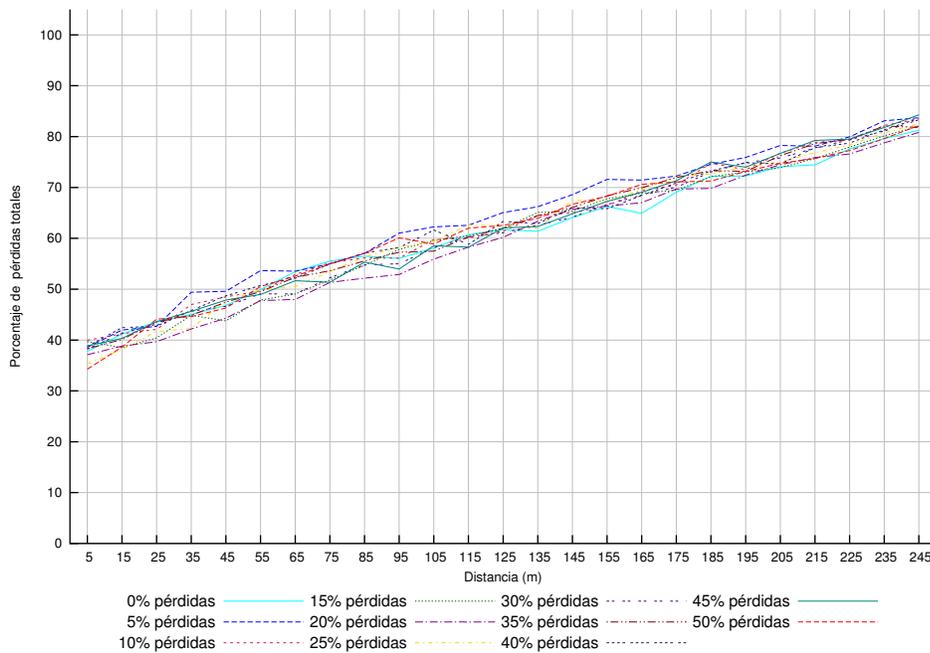


Figura 5.2: Porcentaje de pérdidas escenario 1. Encaminamiento usando AODV, con un rango de transmisión de 15 m.

En un principio se puede observar que el protocolo desarrollado en esta tesis no representa una mejora a la hora de entregar paquetes, pero haciendo un análisis más a fondo de los resultados, podemos darnos cuenta que en realidad el protocolo AODV no realiza una entrega eficiente de los paquetes. El protocolo AODV como se ha mencionado se basa en la recepción de mensajes de acuse para poder mantener sus tablas de rutas actualizadas, lo que provoca que en un ambiente propenso a grandes pérdidas, estos mensajes de acuse tienen la misma probabilidad de perderse. Siendo más específicos, al usar el protocolo AODV se entregan más paquetes de datos que con el protocolo desarrollado. Sin embargo, estos mensajes de datos provienen sólo de algunos nodos de toda la red, es decir no todos los nodos entregan mensajes de datos a la estación base, caso contrario con lo que ocurre al usar nuestro protocolo en donde aunque el porcentaje de mensajes entregados es menor, se garantiza que todos los nodos de la red entregan por lo menos un mensaje a la estación base.

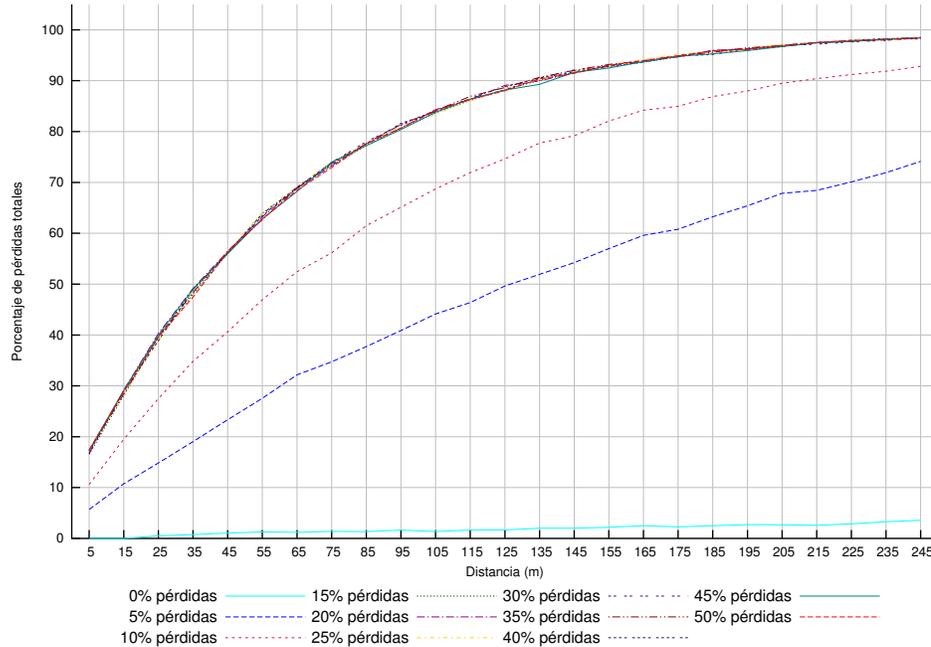


Figura 5.3: Porcentaje de pérdidas escenario 3. Encaminamiento usando EMT, con un rango de transmisión de 15 m.

Recordemos que los escenarios 1 y 3 representan un caso en el que el rango de transmisión de los nodos es de 15 metros, lo cual representa que existe más posibilidad de pérdidas de paquetes, ya que para lograr que un mensaje llegue a la estación base, este debe ser encaminado por más nodos. En las figuras 5.4 y 5.5, tenemos el caso en que el rango de transmisión de los nodos es de 40 metros, con lo cual un paquete proveniente de algún nodo en la red tiene que dar menos saltos para llegar a la estación base, con lo cual se disminuye la posibilidad de pérdidas de paquetes.

Se puede observar que en los escenarios 2 y 4, nuestro protocolo se comporta de manera similar a AODV, incluso podemos notar un porcentaje de pérdidas menor, además del hecho que usando AODV no todos los nodos de la red entregan paquetes de datos a la estación base.

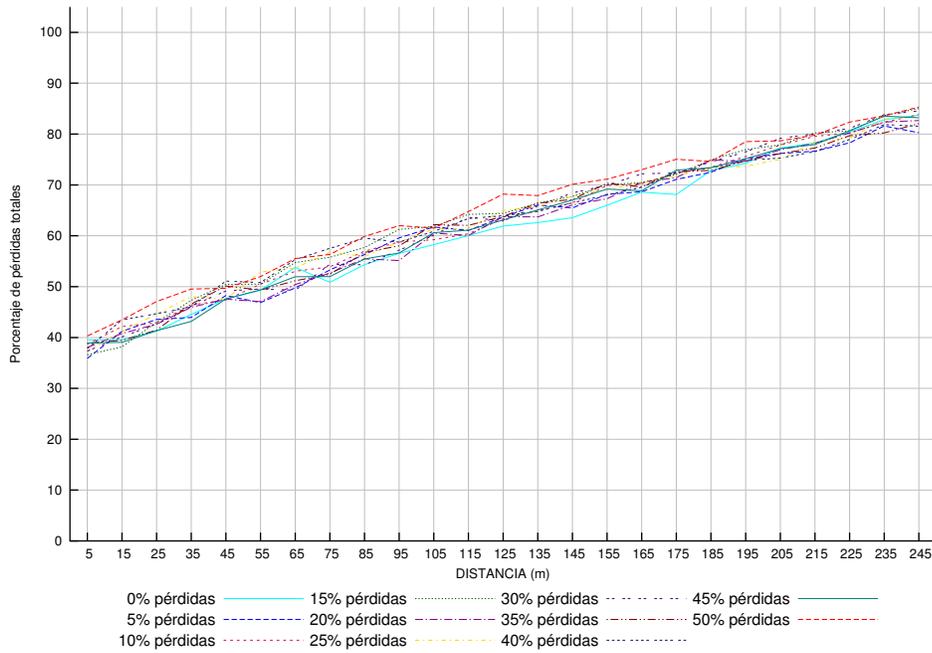


Figura 5.4: Porcentaje de pérdidas escenario 2. Encaminamiento usando AODV, con un rango de transmisión de 40 m.

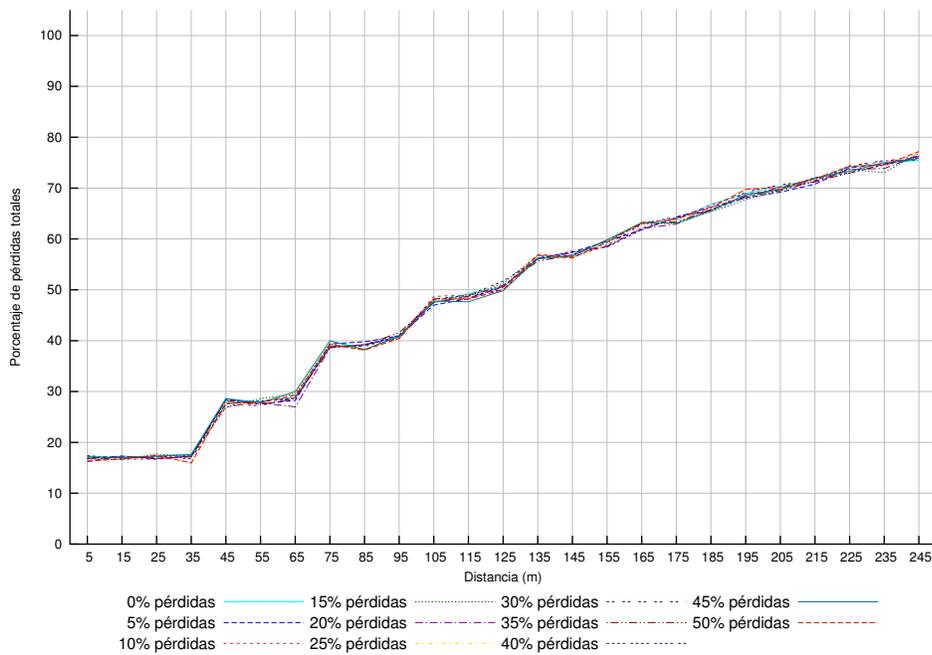


Figura 5.5: Porcentaje de pérdidas escenario 4. Encaminamiento usando EMT, con un rango de transmisión de 40 m.

5.3. Tiempo de entrega de paquetes

Para poder evaluar de una mejor manera cómo es que se comportan los protocolos estudiados, se encontró el tiempo promedio máximo, si es que existe durante la simulación, en la que todos los nodos de la red entregan su primer paquete a la estación base, la simulación dura 503 segundos por lo que en las gráficas siguientes un número mayor a 503 segundos significa que no todos los nodos entregaron por lo menos un paquete durante el tiempo de simulación.

Las figuras 5.6 y 5.7 presentan los tiempos de entrega de los protocolos AODV y el desarrollado en esta tesis, respectivamente, con un rango de transmisión de los nodos de 15 metros. Como podemos observar en el caso del escenario 1, el promedio de las 100 simulaciones nos muestra que usando el protocolo AODV no se recibió por lo menos un mensaje de cada nodo en la red, en la estación base.

Usando nuestro protocolo podemos ver una mejora bajo las mismas condiciones (escenario 3). En la Figura 5.7 podemos observar que con un porcentaje de pérdidas menor a 10% se logra la entrega de por lo menos un paquete de cada nodo en el tiempo de simulación.

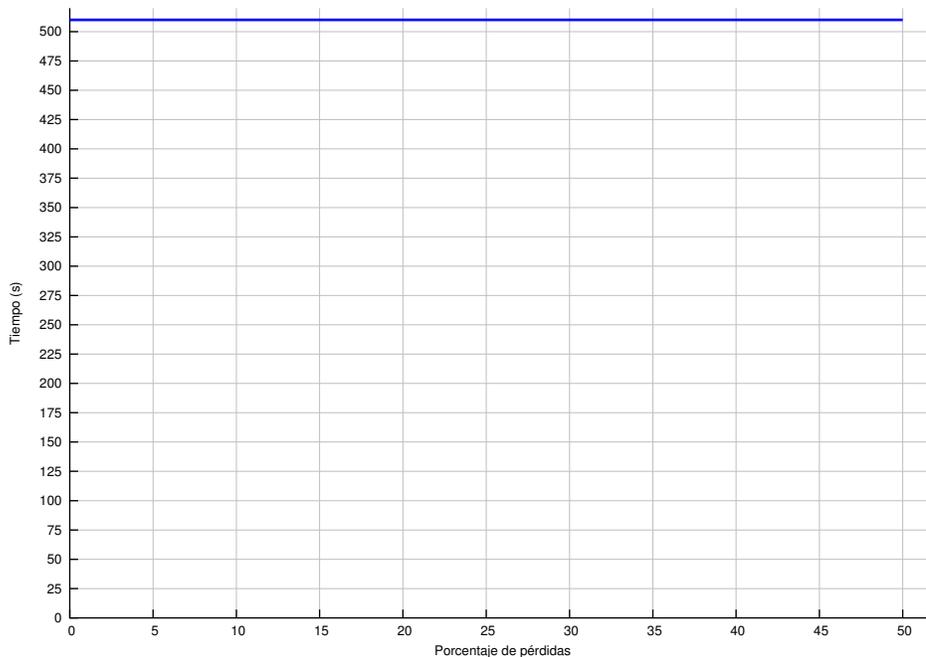


Figura 5.6: Tiempo de entrega de paquetes escenario 1. Encaminamiento usando AODV, con un rango de transmisión de 15 m.

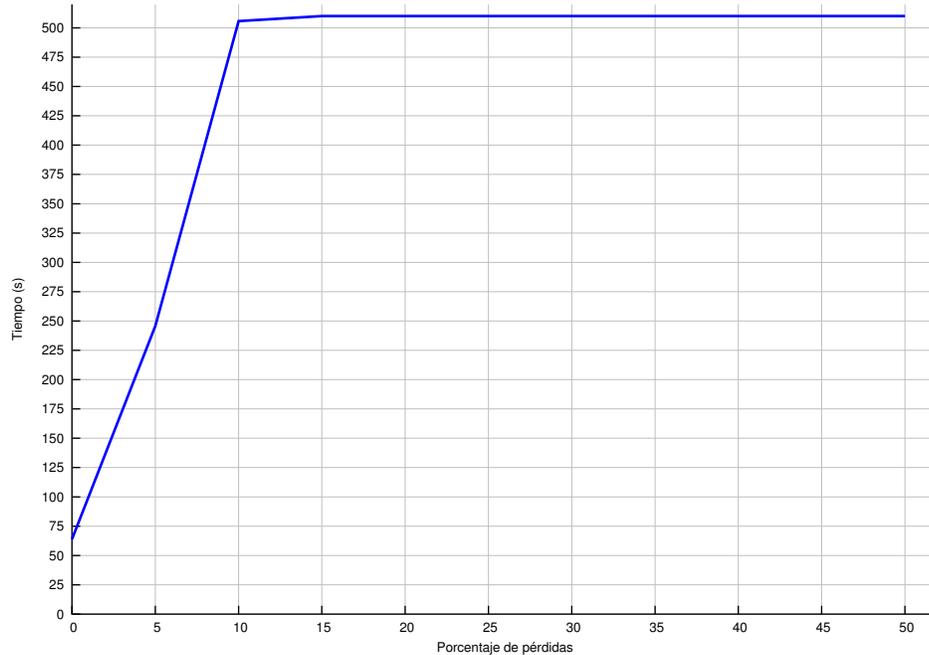


Figura 5.7: Tiempo de entrega de paquetes escenario 3. Encaminamiento usando EMT, con un rango de transmisión de 15 m.

En las Figuras 5.8 y 5.9 se muestra el comportamiento de ambos protocolos cuando el rango de transmisión de los nodos es de 40 metros. En este caso podemos ver que el protocolo AODV (Figura 5.4), no logra entregar un paquete de cada nodo en el tiempo de simulación. En cambio, podemos observar que nuestro protocolo logra la entrega de por lo menos un paquete por cada nodo de la red en un tiempo constante entre 260 y 290 segundos, sin importar el porcentaje de pérdidas de paquetes en la red.

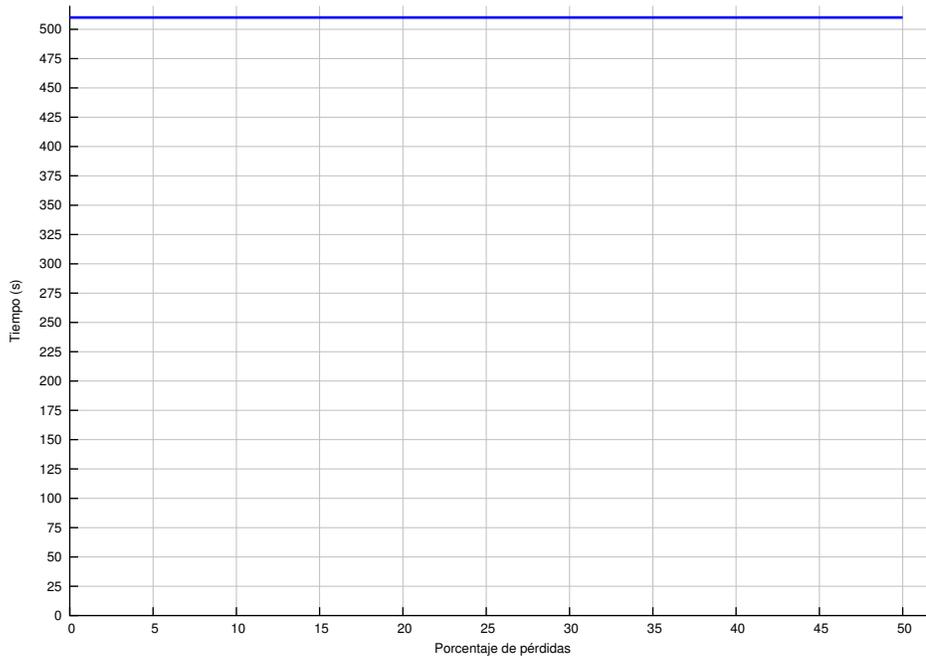


Figura 5.8: Tiempo de entrega de paquetes escenario 2. Encaminamiento usando AODV, con un rango de transmisión de 40 m.

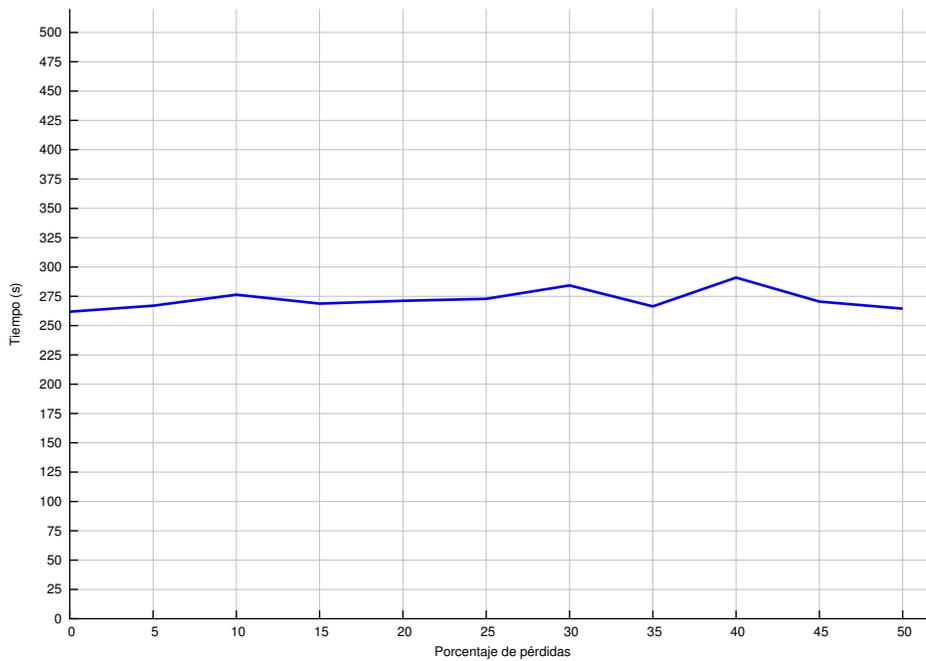


Figura 5.9: Tiempo de entrega de paquetes escenario 4. Encaminamiento usando EMT, con un rango de transmisión de 40 m.

5.4. Pruebas con la implementación de la red de sensores

Con la implementación de la red de sensores, en un prototipo físico, observamos el comportamiento de los protocolos, así como los problemas a los que se enfrenta el sistema. Uno de obstáculos que se observaron en ambientes con ruido, es la sincronía en los mensajes, ya que a causa del ruido es difícil recibir mensajes sin errores que puedan ser interpretados de manera correcta. Con la codificación Manchester, logramos tener una mejor sincronía al leer los mensajes recibidos y de esta forma tener una mayor certeza de que el mensaje recibido es correcto.

Además con el prototipo de la red, podemos probar la etapa de detección de errores. Esta detección se logra mediante el campo de suma de verificación de los mensajes recibidos, permitiendo desechar aquellos mensajes que sean detectados como erróneos. En un ambiente con ruido es difícil mantener todos los mensajes correctos, por tal motivo es importante tener un mecanismo que nos permita eliminar aquellos mensajes corruptos a causa del ruido.

Conclusiones y recomendaciones para trabajo futuro

Las redes de sensores inalámbricas representan una gran herramienta para la monitorización de eventos en un gran número de escenarios, sin embargo no se pueden generalizar los protocolos de encaminamiento y acceso al medio. En efecto, para que las redes de sensores inalámbricas operen en todos los escenarios, es indispensable contar con protocolos de comunicación que se adapten a cada aplicación que se quiere desarrollar. Uno de los campos menos estudiados son los escenarios en ambientes con ruido, en donde los protocolos de comunicación usados comúnmente en las redes de sensores, presentan diversos problemas que conllevan al mal funcionamiento de la red de sensores.

En este trabajo de investigación se logra diseñar e implementar un protocolo de comunicación para una red de sensores inalámbrica, capaz de operar en ambientes con ruido. Esto se logra mediante la implementación de un protocolo de encaminamiento de datos, que no depende de mensajes de acuse para su operación. Además de un protocolo de acceso al medio que pertenece a la familia de protocolos conocidos como TDMA. Por último el mecanismo de detección de errores, mediante la comprobación del campo de suma de verificación de los mensajes, es posible identificar los mensajes corruptos a causa del ruido y de esta forma desecharlos.

Con los resultados obtenidos de la simulación de los protocolos de encaminamiento y acceso al medio, podemos observar que se puede tener una red de sensores inalámbrica capaz de operar en ambientes con ruido, en donde la prioridad es que todos los nodos de la red entreguen por lo menos un mensaje en un tiempo máximo. Esto se logra gracias a que el protocolo de encaminamiento, no usa mensajes de acuse para su operación por lo que las tablas de rutas pueden ser generadas de forma más sencilla, en comparación con otros

protocolos de encaminamiento. Por su parte el protocolo de acceso al medio provee una manera ordenada y sencilla en que los nodos pueden acceder al canal de comunicación, evitando las interferencias con los nodos vecinos.

No obstante, aún queda trabajo por realizar, como el lograr que la red sea autoconfigurable en todos los aspectos o mejorar los tiempos de entrega de paquetes a la estación base. Teniendo un mecanismo de detección de errores, se puede buscar la manera de corregir algunos de los errores que se presenten en los mensajes a causa del ruido. Otra manera en que la red de sensores puede operar de forma más eficiente es aprovechar al máximo las rebanadas de tiempo, para ello se puede en un futuro, contar con un mecanismo, que de forma similar a las redes cognitivas, sea capaz de encontrar las rebanadas libres en un vecindario de nodos. Además es importante realizar un estudio del balance de carga de los nodos para logara un mejor desempeño.

Referencias

- [1] R. Fernández Martínez, et al, “*Redes inalámbricas de sensores: teoría y aplicación práctica*”, Publicaciones Universidad de la Rioja, 2009
- [2] R. Hidalgo Lopez and J. I. Moreno Novella, “*Routing Design in Wireless Sensor Networks and a Solution for Healthcare Environments*”, IEEE Lat. Am. Trans., 2011.
- [3] J. C. Reyes Guerrero, “*Localización en interiores usando redes inalámbricas de sensores: sensores estudio teórico*”, Proyecto de final de carrera, Universidad de Sevilla.
- [4] S. K. Singh, M. . Singh, and D. K. Singh, “*Routing Protocols in Wireless Sensor Networks - A Survey*”, Int. J. Comput. Sci. Eng. Surv., 2010.
- [5] K. Langendoen, “*Medium Access Control in Wireless Networks*“, vol. II. Nova Science Publishers, Inc., 2008.
- [6] M. C. Vuran and I. F. Akyildiz, “*Error Control in Wireless Sensor Networks: A Cross Layer Analysis*”, IEEE/ACM Trans. Netw., 2009.
- [7] C. E. Perkins and E. M. Royer, “*Ad-hoc on-demand distance vector routing*”, Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, 1999.
- [8] C. Perkins, “*Ad hoc On-Demand Distance Vector (AODV) Routing*”, RFC 3561, 2003.
- [9] The Network Simulator ns-2. <http://www.isi.edu/nsnam/ns/>, 2013.
- [10] H. Lee, A. Cerpa, and P. Levis, “*Improving wireless simulation through noise modeling*”, Proc. 6th Int. Conf. Inf. Process. Sens. networks - IPSN '07, 2007.
- [11] M. Moeneclaey, “*General Class of PCM Formats, Including*”, 1984.
- [12] Microchip Technology, “*PIC18F26J50 Family Data Sheet*”, 2011.
- [13] Linx, “*LR Series Transmitter Module Data Guide*”, Data Sheet, 2012.

-
- [14] Linx, “*LR Series Receiver Module Data Guide*”, Data Sheet, 2012.
- [15] MikroC for PIC. <http://www.mikroe.com/mikroc/pic/>, 2013.
- [16] NS Documentation, <http://www.isi.edu/nsnam/ns/doc/>, 2013.
- [17] D. M. Archila Córdoba and F. A. Santamaría Buitrago, “*Estado del arte de las redes de sensores inalámbricos*”, Revista Digital TIA, 2013.
- [18] J. Cayani, E. San Román, J. Chilo y P. Stenumgaard, *Caracterización de Ambientes Industriales Para Aplicaciones de Redes de Sensores Inalámbricos en la Región Arequipa*, CIP CDA CIE ELECTRO I+D, 2012.
- [19] Iacono, P. Godoy, O. Marianetti, C. G. Garino, and C. Párraga, “*Estudio de la Integración entre WSN y redes TCP / IP A Survey on the integration between WSN and TCP / IP networks*”, 2012.
- [20] T. M. Rahayu, S.-G. Lee, and H. Lee, “*Survey on LEACH-based security protocols*”, in 16th International Conference on Advanced Communication Technology, 2014.
- [21] R. Yadav, S. Varma, and N. Malaviya, “*A survey of mac protocols for wireless sensor networks*”, UbiCC J., 2009.
- [22] D. Puccinelli and M. Haenggi, “*Wireless sensor networks: applications and challenges of ubiquitous sensin*”, IEEE Circuits Syst. Mag., 2005.
- [23] A. Hussain, “*WSN Research Activities for Military Application*” Adv. Commun. Technol., 2009.
- [24] S. Giannoulis, a. Prayati, C. Antonopoulos, and G. Papadopoulos, “*A.MO.R : An adaptive routing mechanism for WSN health applications*”, Proc. 3d Int. ICST Conf. Pervasive Comput. Technol. Healthc., 2009.
- [25] J. Kenyeres, Š. Šajban, P. Farkaš, and M. Rakús, “*Indoor experiment with WSN application*”, MIPRO, 2010.
- [26] A. F. Murillo, M. Pena, and D. Martinez, “*Applications of WSN in health and agriculture*”, IEEE Colombian Communications Conference (COLCOM), 2012.
- [27] Q. Chi, H. Yan, C. Zhang, Z. Pang, L. Da Xu, and S. Member, “*A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment*”, IEEE Trans. Ind. Informatics, 2014.
-

Apéndices

Apéndice A

Script TCL

```
1 # =====
2 # Definicion de parametros
3 # =====
4 set val(chan) Channel/WirelessChannel ;# Tipo de canal
5 set val(prop) Propagation/TwoRayGround ;# Modelo de propagacion
6 set val(netif) Phy/WirelessPhy/802_15_4 ;# Interfaz de red
7 set val(mac) Mac/802_15_4 ;# Tipo Acceso al medio
8 set val(ifq) Queue/DropTail/PriQueue ;# Tipo de cola
9 set val(ll) LL ;# Tipo de enlace
10 set val(ant) Antenna/OmniAntenna ;# Antena
11 set val(ifqlen) 10 ;# Numero maximo de paquetes en cola
12 set val(nn) 101 ;# Numero de nodos + estacion base
13 set val(rp) WFRP ;# Protocolo de enrutamiento
14 set val(x) 510 ;# Dimension X del escenario (metros)
15 set val(y) 30 ;# Dimension Y del escenario (metros)
16 set val(traffic) cbr ;# Tipo de trafico
17
18 set ns [new Simulator]
19 set tracefd [open 100.tr w] ;# Archivo de trazas
20 set namtrace [open 100.nam w] ;# Archivo nam
21 set stopTime 503 ;# En segundos
22
23 $ns trace-all $tracefd
24 $ns namtrace-all-wireless $namtrace $val(x) $val(y)
25
26 $ns puts-nam-traceall {# nam4wpn #}
27
28 Mac/802_15_4 wpanNam namStatus on
29
30 # =====
31 # Modelo TwoRayGround
32 # =====
33
34 set dist(5m) 7.69113e-06
35 set dist(9m) 2.37381e-06
36 set dist(10m) 1.92278e-06
37 set dist(11m) 1.58908e-06
38 set dist(12m) 1.33527e-06
```

```
39 set dist(13m) 1.13774e-06
40 set dist(14m) 9.81011e-07
41 set dist(15m) 8.54570e-07
42 set dist(16m) 7.51087e-07
43 set dist(20m) 4.80696e-07
44 set dist(25m) 3.07645e-07
45 set dist(30m) 2.13643e-07
46 set dist(35m) 1.56962e-07
47 set dist(40m) 1.20174e-07
48 Phy/WirelessPhy set CStresh_ $dist(40m)
49 Phy/WirelessPhy set RXThresh_ $dist(40m)
50
51 # Craecion del escenario
52 set topo [new Topography]
53 $topo load_flatgrid $val(x) $val(y)
54
55 create-god $val(nn)
56
57 # =====
58 # Configuracion de los nodos
59 # =====
60
61 $ns node-config -adhocRouting $val(rp) \
62                 -llType $val(ll) \
63                 -macType $val(mac) \
64                 -ifqType $val(ifq) \
65                 -ifqLen $val(ifqlen) \
66                 -antType $val(ant) \
67                 -propType $val(prop) \
68                 -phyType $val(netif) \
69                 -channel [new $val(chan)] \
70                 -topoInstance $topo \
71                 -agentTrace ON \
72                 -routerTrace ON \
73                 -macTrace ON \
74                 -movementTrace OFF
75
76 for {set i 0} {$i < $val(nn)} { incr i } {
77     set node_($i) [$ns node]
78     $node_($i) random-motion 0
79 }
80
81 # Se carga el archivo con la posicion de los nodos
82 source ./E100_10.scn
83
84 # Posicion de la estacion base
85 $node_(0) set X_ [ expr {$val(x)/2} ]
86 $node_(0) set Y_ [ expr {$val(y)/2} ]
87 $node_(0) set Z_ 0.0
88 $node_(0) label "Estacion Base"
89
90 # Se envia el primer mensaje de descubrimiento de la estacion base
91 $ns at 1.0 "[$node_(0) set ragent_] sink"
92
93 for {set i 0} {$i < $val(nn)} { incr i } {
94     $ns initial_node_pos $node_($i) 2
95 }
96
```

```

97 # Procedimiento para generar trafico CBR sobre UDP
98 proc cbrtraffic { src dst interval starttime } {
99     global ns node_
100    set udp($src) [new Agent/UDP]
101    eval $ns attach-agent \ $node_($src) \ $udp($src)
102    set null($dst) [new Agent/Null]
103    eval $ns attach-agent \ $node_($dst) \ $null($dst)
104    set cbr($src) [new Application/Traffic/CBR]
105    eval \ $cbr($src) set packetSize_ 14
106    eval \ $cbr($src) set interval_ $interval
107    eval \ $cbr($src) set random_ 0
108    eval \ $cbr($src) set maxpkts_ 1
109    eval \ $cbr($src) attach-agent \ $udp($src)
110    eval $ns connect \ $udp($src) \ $null($dst)
111    $ns at $starttime "$cbr($src) start"
112 }
113
114 global ns node_
115 Mac/802_15_4 wpanCmd ack4data off
116 Mac/802_15_4 wpanNam PlaybackRate 0.1
117 set tiempo 2.0 ; # En este tiempo iniciara la transmision de paquetes
118 # Rutina que simula el protocolo de acceso al medio (ranuras de 200 ms)
119 while {$tiempo < [expr {$stoptime - 5}]} {
120     for {set nodo 1} {$nodo < $val(nn)} {incr nodo} {
121         set tiempo [expr {$tiempo + 0.2}]
122         eval $val(traffic)traffic $nodo 0 1 $tiempo
123         set pktType cbr
124         Mac/802_15_4 wpanNam FlowClr -p $pktType -s $nodo -d 0 -c blue
125         $ns at $tiempo "$node_($nodo) NodeClr blue"
126         $ns at $tiempo "$node_(0) NodeClr green4"
127         $ns at $tiempo "$ns trace-annotate \"(at $tiempo) $val(traffic) traffic from node $nodo to node
128         0\""
129     }
130 }
131 # Reset de los nodos al terminar el tiempo de simulacion
132 for {set i 0} {$i < $val(nn)} {incr i} {
133     $ns at $stoptime "$node_($i) reset;"
134 }
135
136 # Fin de la simulacion NAM
137 $ns at $stoptime "$ns nam-end-wireless $stoptime"
138 $ns at $stoptime "stop"
139 $ns at [expr $stoptime + 0.01] "puts \"end simulation\"; $ns halt"
140
141 proc stop {} {
142     global ns tracefd namtrace
143     $ns flush-trace
144     close $tracefd
145     close $namtrace
146     # exec nam 100.nam &
147     exit 0
148 }
149
150 $defaultRNG seed 0 ;# Semilla aleatoria para diferentes resultados en cada simulacion
151
152 $ns run

```

Apéndice B

Scripts de filtrado de datos

```
1  #!/bin/csh
2
3  set NUM_NODOS = 100
4  set SEPARACION = 10
5  set NUM_SIMULACIONES = 100
6  set PORCENTAJE_PERDIDAS = 0
7
8  set NUM_PRUEBA = 0
9  set P_ENV = 0
10 set P_REC = 0
11 set TP_ENV = 0
12 set TP_REC = 0
13 set PERDIDAS = 0
14 set T_PERDIDAS = 0
15 set AUX = 0
16 set AUX1 = 0
17 set AUX2 = 0
18 set DIST = 0
19 set DIST1 = 0
20 set DIST2 = 0
21 set COOR_EB = 0
22 set TIEMPO = 0
23 set NL = 0
24 set CONTADOR = 0
25 set ENV = 0
26 set REC = 0
27 set PER = 0
28 set PROM_ENV = 0
29 set PROM_REC = 0
30 set PROM_PER = 0
31 set PORC_PERDIDAS = 0
32
33 set DIR_T = ./TEMP
34 set DIR_R = ./RESULTADOS
35 set DIR_N = /media/MIKTLAN/UAM-I/NS-Simulacion/NS/AODV_40MTS
36
37
38
```

```

39 foreach PP ( 5 10 15 20 25 30 35 40 45 50 )
40 # foreach PP ( 0 5 10 15 20 25 30 35 40 45 50 )
41 rm $DIR_T/*
42 @ PORCENTAJE_PERDIDAS = $PP
43 set NAME_DIR = {$PORCENTAJE_PERDIDAS}_perdidas
44 @ NUM_PRUEBA = 1
45 while ( $NUM_PRUEBA <= $NUM_SIMULACIONES)
46   set NAME_ARCH = {$NUM_NODOS}_{$PORCENTAJE_PERDIDAS}_{$NUM_PRUEBA}
47   echo " "
48   echo PRUEBA $NUM_PRUEBA de $NUM_SIMULACIONES
49   echo PORCENTAJE DE PERDIDAS $PORCENTAJE_PERDIDAS
50   echo " "
51
52   # Se calcula en numero de paquetes enviados, recibidos y perdidos en la simulacion
53   awk ' $1 ~ /s/ && $4 ~ /AGT/' $DIR_N/$NAME_DIR/$NAME_ARCH.tr > $DIR_T/temp #filtrado de los
      mensajes de datos enviados
54   @ P_ENV = `wc -l $DIR_T/temp | cut -d " " -f1`
55   @ TP_ENV = $TP_ENV + $P_ENV
56   rm $DIR_T/temp
57
58   awk '$1 ~ /r/ && $4 ~ /AGT/' $DIR_N/$NAME_DIR/$NAME_ARCH.tr > $DIR_T/temp #filtrado de les
      mensajes de datos recibidos
59   @ P_REC = `wc -l $DIR_T/temp | cut -d " " -f1`
60   @ TP_REC = $TP_REC + $P_REC
61   rm $DIR_T/temp
62
63   @ PERDIDAS = $P_ENV - $P_REC
64   @ T_PERDIDAS = $T_PERDIDAS + $PERDIDAS
65
66   # Se calcula el tiempo que tardan todos los nodos en entregar su primer paquete
67   set TIEMPO = 0
68   awk '{ if ((index($1,"r")) && (index($7,"cbr")==1) && (index($4,"AGT")==1)) {print $2 "\t" $14}
      }' $DIR_N/$NAME_DIR/$NAME_ARCH.tr > $DIR_T/temp
69   tr -d '[' < $DIR_T/temp > $DIR_T/temp1
70   awk -F " " '{print $1}' $DIR_T/temp1 > $DIR_T/temp2
71   awk '{print $1"\t"$2}' $DIR_T/temp2 > $DIR_T/temp3
72
73   set AUX = 1
74   while ($AUX <= $NUM_NODOS )
75     cat $DIR_T/temp3 | grep "_${AUX}_ " | sed q >> $DIR_T/temp4
76     @ AUX ++
77   end
78   awk '{print $1}' $DIR_T/temp4 >> $DIR_T/temp5
79
80   @ AUX1 = `wc -l $DIR_T/temp5 | cut -d " " -f1`
81
82   if ($AUX1 < $NUM_NODOS) then
83     @ TIEMPO = 510
84   else
85     foreach line ("cat $DIR_T/temp5")
86       set AUX2 = `echo "($line > $TIEMPO)" | bc -l`
87       if ($AUX2 == 1) then
88         set TIEMPO = `echo "$line" | bc -l`
89       endif
90     end
91   endif
92
93   echo $TIEMPO >> $DIR_T/tiempos_$PORCENTAJE_PERDIDAS

```

```

94  rm $DIR_T/temp*
95
96  echo "PRUEBA $NUM_PRUEBA de $NUM_SIMULACIONES">> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
97  echo "PAQUETES ENVIADOS:      $P_ENV" >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
98  echo "PAQUETES ENTREGADOS:    $P_REC" >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
99  echo "PAQUETES PERDIDOS:     $PERDIDAS" >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
100 echo "TOTAL PAQUETES ENVIADOS:  $TP_ENV" >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
101 echo "TOTAL PAQUETES ENTREGADOS: $TP_REC" >>$DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
102 echo "TOTAL PAQUETES PERDIDOS:  $T_PERDIDAS" >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
103 echo "TIEMPO PROMEDIO DE ENTREGA DE PAQUETES:  $TIEMPO" >>
    $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
104 echo " " >> $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS
105
106 echo " "
107 echo "PAQUETES ENVIADOS:      $P_ENV"
108 echo "PAQUETES ENTREGADOS:    $P_REC"
109 echo "PAQUETES PERDIDOS:     $PERDIDAS"
110 echo "TOTAL PAQUETES ENVIADOS:  $TP_ENV"
111 echo "TOTAL PAQUETES ENTREGADOS: $TP_REC"
112 echo "TOTAL PAQUETES PERDIDOS:  $T_PERDIDAS"
113 echo "TIEMPO PROMEDIO DE ENTREGA DE PAQUETES:  $TIEMPO"
114 echo " "
115 echo " "
116
117 set AUX = 1
118 @ COOR_EB = (($SEPARACION * $NUM_NODOS) + (2 * $SEPARACION)) / 4
119 @ DIST1 = $COOR_EB + ($SEPARACION / 2)
120 @ DIST2 = $COOR_EB - ($SEPARACION / 2)
121
122 @ DIST = $COOR_EB - $DIST2
123 echo $DIST >> $DIR_T/dist_$DIST{m}
124 echo $DIST >> $DIR_T/dist_$DIST{m}
125 echo $DIST >> $DIR_T/dist_$DIST{m}
126 echo $DIST >> $DIR_T/dist_$DIST{m} #Se guarda la distancia
127
128 cat $DIR_N/$NAME_DIR/orden_nodos_{$PORCENTAJE_PERDIDAS}_{$NUM_PRUEBA} | grep "_${DIST1}" >
    $DIR_T/temp1
129 cat $DIR_N/$NAME_DIR/orden_nodos_{$PORCENTAJE_PERDIDAS}_{$NUM_PRUEBA} | grep "_${DIST2}" >>
    $DIR_T/temp1 #Nodos y distancia
130 cut -d " " -f1 $DIR_T/temp1 > $DIR_T/temp2 #Solo nodos
131 cut -d " " -f1 $DIR_T/temp1 >> $DIR_T/nodos_$DIST{m} #Solo nodos
132
133 foreach line ("cat $DIR_T/temp2")
134     cat $DIR_N/$NAME_DIR/$NAME_ARCH.tr | grep "_${line}" > $DIR_T/temp3
135     awk '$1 ~ /s/ && $7 ~ /cbr/ && $4 ~ /AGT/' $DIR_T/temp3 > $DIR_T/temp4
136     wc -l $DIR_T/temp4 | cut -d " " -f1 >> $DIR_T/enviados_$DIST{m}
137 end
138
139 awk '{ if ((index($1,"r")) && (index($7,"cbr")==1) && (index($4,"AGT")==1)) {print $14} }'
    $DIR_N/$NAME_DIR/$NAME_ARCH.tr > $DIR_T/temp5
140 tr -d '[' < $DIR_T/temp5 > $DIR_T/temp6
141 awk -F " " '{print "_"$1"}' $DIR_T/temp6 > $DIR_T/recibidos
142
143 foreach line ("cat $DIR_T/temp2")
144     cat $DIR_T/recibidos | grep "_${line}" > $DIR_T/temp7
145     wc -l $DIR_T/temp7 | cut -d " " -f1 >> $DIR_T/recibidos_$DIST{m}
146 end
147

```

```

148  paste $DIR_T/enviados_${DIST}{m} $DIR_T/recibidos_${DIST}{m} > $DIR_T/temp8
149  awk '{perdidas=0; for (i=1; i<=1; i++) perdidas=$i-$(i+1); print perdidas}' $DIR_T/temp8 >
    $DIR_T/perdidas_${DIST}{m}
150
151
152  rm $DIR_T/temp*
153
154  @ AUX++
155
156  while ($AUX <= $NUM_NODOS / 4)
157    @ DIST1 = $DIST1 + $SEPARACION
158    @ DIST2 = $DIST2 - $SEPARACION
159    @ DIST = $COOR_EB - $DIST2
160
161    echo $DIST >> $DIR_T/dist_${DIST}{m}
162    echo $DIST >> $DIR_T/dist_${DIST}{m}
163    echo $DIST >> $DIR_T/dist_${DIST}{m}
164    echo $DIST >> $DIR_T/dist_${DIST}{m} #Se guarda la distancia
165
166    cat $DIR_N/$NAME_DIR/orden_nodos_{$PORCENTAJE_PERDIDAS}_{$NUM_PRUEBA} | grep "_${DIST1}_" >
    $DIR_T/temp1
167    cat $DIR_N/$NAME_DIR/orden_nodos_{$PORCENTAJE_PERDIDAS}_{$NUM_PRUEBA} | grep "_${DIST2}_" >>
    $DIR_T/temp1 #Nodos y distancia
168    cut -d " " -f1 $DIR_T/temp1 > $DIR_T/temp2 #Solo nodos
169    cut -d " " -f1 $DIR_T/temp1 >> $DIR_T/nodos_${DIST}{m} #Solo nodos
170
171    foreach line ("cat $DIR_T/temp2")
172      cat $DIR_N/$NAME_DIR/$NAME_ARCH.tr | grep "_${line}_" > $DIR_T/temp3
173      awk '$1 ~ /s/ && $7 ~ /cbr/ && $4 ~ /AGT/' $DIR_T/temp3 > $DIR_T/temp4
174      wc -l $DIR_T/temp4 | cut -d " " -f1 >> $DIR_T/enviados_${DIST}{m}
175    end
176
177    awk '{ if ((index($1,"r")) && (index($7,"cbr")==1) && (index($4,"AGT")==1)) { print $14 } }'
    $DIR_N/$NAME_DIR/$NAME_ARCH.tr > $DIR_T/temp5
178    tr -d '[' < $DIR_T/temp5 > $DIR_T/temp6
179    awk -F ":" '{print "_"$1"}' $DIR_T/temp6 > $DIR_T/recibidos
180
181    foreach line ("cat $DIR_T/temp2")
182      cat $DIR_T/recibidos | grep "_${line}_" > $DIR_T/temp7
183      wc -l $DIR_T/temp7 | cut -d " " -f1 >> $DIR_T/recibidos_${DIST}{m}
184    end
185
186    paste $DIR_T/enviados_${DIST}{m} $DIR_T/recibidos_${DIST}{m} > $DIR_T/temp8
187    awk '{perdidas=0; for (i=1; i<=1; i++) perdidas=$i-$(i+1); print perdidas}' $DIR_T/temp8 >
    $DIR_T/perdidas_${DIST}{m}
188
189    @ AUX++
190    rm $DIR_T/temp*
191  end
192
193  @ NUM_PRUEBA = $NUM_PRUEBA + 1
194
195  end
196
197  echo " "
198  echo "Iniciando procesamiento de datos para $PORCENTAJE_PERDIDAS % de perdidas"
199
200  @ DIST1 = $COOR_EB + ($SEPARACION / 2)

```

```

201 @ DIST2 = $COOR_EB - ($SEPARACION / 2)
202 @ DIST = $COOR_EB - $DIST2
203
204 @ NL = `wc -l $DIR_T/nodos_$DIST{m} | cut -d " " -f1 `
205
206 @ ENV = 0
207 foreach line (`cat $DIR_T/enviados_$DIST{m}`)
208   set ENV = `echo "($line + $ENV)" | bc -l `
209 end
210 set PROM_ENV = `echo "scale=5;($ENV / $NL)" | bc -l `
211 @ REC = 0
212 foreach line (`cat $DIR_T/recibidos_$DIST{m}`)
213   set REC = `echo "($line + $REC)" | bc -l `
214 end
215 set PROM_REC = `echo "scale=5;($REC / $NL)" | bc -l `
216 @ PER = 0
217 foreach line (`cat $DIR_T/perdidas_$DIST{m}`)
218   set PER = `echo "($line + $PER)" | bc -l `
219 end
220 set PROM_PER = `echo "scale=5;($PER / $NL)" | bc -l `
221 set PORC_PERDIDAS = `echo "scale=5;(100 * ($PROM_PER) / $PROM_ENV)" | bc -l `
222 echo "$DIST $ENV $REC $PER $PROM_ENV $PROM_REC $PROM_PER $PORC_PERDIDAS" >>
    $DIR_T/resultados_$PORCENTAJE_PERDIDAS
223 set AUX = 2
224 while ($AUX <= $NUM_NODOS / 4)
225   @ DIST1 = $DIST1 + $SEPARACION
226   @ DIST2 = $DIST2 - $SEPARACION
227   @ DIST = $COOR_EB - $DIST2
228
229   @ NL = `wc -l $DIR_T/nodos_$DIST{m} | cut -d " " -f1 `
230
231   @ ENV = 0
232   foreach line (`cat $DIR_T/enviados_$DIST{m}`)
233     set ENV = `echo "($line + $ENV)" | bc -l `
234   end
235   set PROM_ENV = `echo "scale=5;($ENV / $NL)" | bc -l `
236   @ REC = 0
237   foreach line (`cat $DIR_T/recibidos_$DIST{m}`)
238     set REC = `echo "($line + $REC)" | bc -l `
239   end
240   set PROM_REC = `echo "scale=5;($REC / $NL)" | bc -l `
241   @ PER = 0
242   foreach line (`cat $DIR_T/perdidas_$DIST{m}`)
243     set PER = `echo "($line + $PER)" | bc -l `
244   end
245   set PROM_PER = `echo "scale=5;($PER / $NL)" | bc -l `
246   set PORC_PERDIDAS = `echo "scale=5;(100 * ($PROM_PER) / $PROM_ENV)" | bc -l `
247   echo "$DIST $ENV $REC $PER $PROM_ENV $PROM_REC $PROM_PER $PORC_PERDIDAS" >>
    $DIR_T/resultados_$PORCENTAJE_PERDIDAS
248   @ AUX++
249
250 end
251 mv $DIR_T/tiempos_$PORCENTAJE_PERDIDAS $DIR_R/
252 mv $DIR_T/estadisticas_$PORCENTAJE_PERDIDAS $DIR_R/
253 mv $DIR_T/resultados_$PORCENTAJE_PERDIDAS $DIR_R/
254
255 mkdir $DIR_R/${PORCENTAJE_PERDIDAS}_Perdidas
256 mv $DIR_T/* $DIR_R/${PORCENTAJE_PERDIDAS}_Perdidas

```

```
257
258 end
259
260 echo " "
261 echo " "
262 echo "PROCESAMIENTO DE DATOS TERMINADO . . . "
263 echo " "
264 echo " "
```

Diagrama del circuito sensor

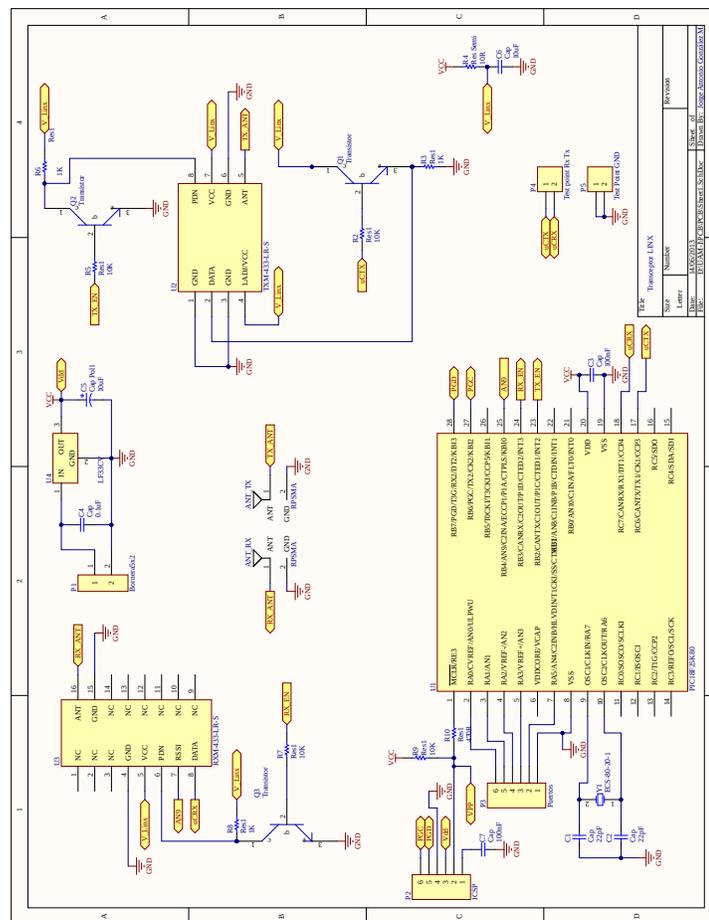


Figura C.1: Diagrama esquemático del sensor