

**Evaluación del Desempeño de la Plataforma de Desarrollo de Redes de
Sensores Inalámbricos Sentilla**

José Carlos Ramírez Ramírez

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO QUIMICAS
ESCUELA DE INGENIERIAS ELECTRICA Y ELECTRONICA
BUCARAMANGA
2009**

**Evaluación del Desempeño de la Plataforma de Desarrollo de Redes de
Sensores Inalámbricos Sentilla**

José Carlos Ramírez Ramírez

**Trabajo de grado presentado como requisito para optar al título de Ingeniero
Electrónico**

Director

Mie. José de Jesús Rúgeles Uribe

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO QUIMICAS
ESCUELA DE INGENIERIAS ELECTRICA Y ELECTRONICA
BUCARAMANGA
2009**

A mi familia.

CONTENIDO

1. INTRODUCCION	1
2. DESCRIPCION DE LAS PLATAFORMAS	1
2.1. Plataformas TelosB y Tmot.....	1
2.2. Plataforma Sentilla	3
2.2.1. Modelo de Programación Sentilla.	4
2.2.2. Características del hardware Sentilla	5
2.3. Plataforma XBee	7
3. PRUEBAS REALIZADAS A LA PLATAFORMA SENTILLA	8
3.1. Alcance y potencia de transmisión de los motes en exteriores	8
3.2. Alcance y potencia de transmisión de los motes en interiores.....	9
3.3. Prueba del protocolos multisalto de diseminación	10
3.4. Exploración de funcionalidades.....	11
4. RESULTADOS.....	12
4.1. Alcance y potencia de transmisión de los motes en exteriores ..	12
4.2. Alcance y potencia de transmisión de los motes en interiores...	13
4.3. Prueba del protocolos multisalto de diseminación	13
4.4. Exploración de funcionalidades.....	16
5. ANALISIS Y DISCUSION.....	16
5.1. Alcance y potencia de transmisión de los motes en exteriores	16
5.2. Alcance y potencia de transmisión de los motes en interiores.....	20
5.3. Prueba del protocolo multisalto de diseminación	20
5.4. Exploración de funcionalidades.....	21
6. AGRADECIMIENTOS.....	23
REFERENCIAS	24

LISTA DE FIGURAS

Figura 1: Motes Tmote y TelosB	2
Figura 2: Diagrama de modelo de programación Sentilla	4
Figura 3: Chip básico de los motes Sentilla.	5
Figura 4: Gateway USB Sentilla	6
Figura 5: Mote JCreate Sentilla.....	6
Figura 6: Accesorio para programación de firmware.....	7
Figura 7: Trama UART del modo API.....	8
Figura 8: Plano del sitio y trayectorias de medida del experimento.....	10
Figura 9: Disposición de los motes para las pruebas de diseminación.....	11
figura 10: valores RSSI tomados por los motes y por el sniffer	12
Figura 11: valores RSSI para cada trayectoria indoor.....	13
Figura 12: Tiempos totales de tránsito en la disposición Y	14
Figura 13: Tiempos de salto relativos para la disposición Y.....	14
Figura 14: Tiempos totales de tránsito en la disposición L.....	15
Figura 15: Tiempos de salto relativos para la disposición L.....	15
Figura 16: Valores RSSI tomados por el Sniffer junto con los del mote corregidos.....	17
Figura 17: Gráficas teórica y práctica para potencia recibida	19

LISTA DE TABLAS

Tabla 1. Conectores para periféricos disponibles en los JCreate.....	6
Tabla 2. : Resumen de resultados: Alcance y potencia de transmisión de los motes en exteriores.....	12
Tabla 3.: promedios relativos de tránsito para la disposición Y.....	16
Tabla 4.: promedios relativos de tránsito para la disposición L.....	16

RESUMEN

TÍTULO: Evaluación del Desempeño de la Plataforma de Desarrollo de Redes de Sensores Inalámbricos Sentilla*

AUTOR: Ramírez Ramírez, José Carlos**

PALABRAS CLAVES: Redes de Sensores inalámbricos, Sentilla, kit Perk, Protocolo de disseminación.

DESCRIPCIÓN:

En este trabajo se evaluaron varias características de la plataforma de desarrollo de redes de sensores inalámbricos Sentilla con el propósito de obtener criterios de comparación respecto a otras plataformas, para esto se utilizó el kit Perk ofrecido por Sentilla para probar la plataforma; Se aplicaron pruebas para observar el alcance y potencia de transmisión variando la distancia entre transmisor y receptor tanto en ambientes exteriores como en interiores utilizando como indicador el valor RSSI de los mensajes enviados por los nodos inalámbricos, se experimentó con el convertidor analógico digital, el módulo UART y los pines digitales de entrada y salida de la plataforma, y se estudió el desempeño de la plataforma en el enrutamiento de mensajes utilizando el protocolo multisalto "dissem" con dos topologías de red distintas.

Como resultado de estas pruebas se encontró que la plataforma posee limitaciones serias en su rango de alcance tanto en interiores como en exteriores, se observó que la velocidad de respuesta de los periféricos se ve afectada al utilizar drivers de alto nivel de abstracción y se observó que para los mensajes difundidos con el protocolo multisalto "dissem" los tiempos de tránsito en cada nodo inalámbrico aumentan con la cantidad de nodos que enrutan sus mensajes a través de él.

*Proyecto de Grado

**Facultad Físico Mecánicas. Escuela Ingenierías Eléctrica y Electrónica. Director: Mie. José de Jesús Rúgeles Uribe.

ABSTRACT

TITLE: EVALUATION OF THE WIRELESS SENSOR NETWORK DEVELOPMENT PLATFORM SENTILLA

AUTHOR: RAMÍREZ RAMÍREZ, José Carlos**

KEY WORDS: Wireless Sensor Network, Sentilla, kit Perk, Dissemination Protocol.

DESCRIPTION:

In this paper were evaluated some features of the wireless sensor network development platform Sentilla with the purpose of obtaining comparative criteria with respect to other platforms, for this we used the Perk evaluation kit offered by Sentilla; Tests to observe the transmission power and range varying the distance between sender and receiver in outdoor and indoor environments were applied using the RSSI value of the messages sent by the wireless nodes as an indicator, we tested also the analog to digital converter, the UART module, the general input/output digital pins, and the platform's performance routing messages using multi-hop protocols.

As a result to this evaluation, serious transmission range limitations were found for both indoor and outdoor environments, we found also that the response speed of the peripherals is affected by using high level drivers and we observed that the transit times at each wireless node for messages sent using the multi-hop protocol "dissem" increased with the amount of nodes routing their messages through it.

*Degree Project

**Fisicomecanicas Faculty. Electric and Electronic Engineering School. Director: Mie. José de Jesús Rúgeles Uribe.

1. INTRODUCCION

Al buscar información acerca del desempeño de la plataforma de desarrollo de sensores inalámbricos Sentilla, debido al carácter propietario de la licencia bajo la que se entrega la misma y a su reciente lanzamiento, no se encuentran referencias serias y/o imparciales, la única información disponible aparte de los manuales y guías entregadas por el fabricante es la información que se ha acumulado en los foros de los usuarios entusiastas que han adquirido la plataforma bajo la licencia de evaluación la cual en contados casos resulta útil pues la mayoría de estos usuarios están interesados en la plataforma mas como pasatiempo que como una herramienta para ámbitos académicos o industriales.

Es de interés evaluar las características de la plataforma Sentilla para obtener información de primera mano que sirva de base para el desarrollo de proyectos académicos y contar con criterios serios respaldados por experiencia de uso, para utilizarlos a la hora de escoger, entre esta y otras dos plataformas disponibles cada una con enfoques y características distintas, cual aplicar en la resolución de problemas para cuya solución sea factible la utilización de redes de sensores inalámbricos.

Para evaluar la plataforma Sentilla se experimento con los dispositivos disponibles en el mote explorando las funcionalidades ofrecidas en las API's respectivas, además de las API's para dispositivos se revisaron y evaluaron las API's de comunicaciones, al hacer esto se logro entendimiento y experiencia en el modelo de programación Sentilla para WSN's (Wireless Sensor Networks); En el caso de los dispositivos la evaluación fue simple y directa, es decir para cada dispositivo se instalaban los programas que utilizaban sus características y se comprobaba su funcionamiento, pero para evaluar las funcionalidades de comunicación y la efectividad del modelo de programación Sentilla se diseñaron y aplicaron pruebas para obtener de manera eficiente información relevante acerca del desempeño de la plataforma.

2. DESCRIPCION DE LAS PLATAFORMAS

Todo el trabajo se realizo sobre la plataforma Sentilla, pero para ofrecer un marco de referencia para los criterios de comparación que se quieren obtener se describirán también las plataformas XBee y TelosB Aunque con menos profundidad.

2.1. Plataformas TelosB y Tmot.

La plataforma TelosB fue desarrollada por la Universidad de Berkeley en compañía con Intel, bajo licencia Open-Source y es la plataforma que utiliza la comunidad científica que trabaja en WSN, la plataforma Tmote es igual a la TelosB excepto por modificaciones en el PCB para reducir el tamaño e integrar un sensor de humedad y luz opcional, era comercializada por Moteiv una empresa cuyo fundador fue uno de los miembros del equipo de Berkeley que desarrolló la plataforma TelosB y que posteriormente transformó Moteiv en Sentilla no sin antes

cambiar la licencia de Open-Source a propietaria.

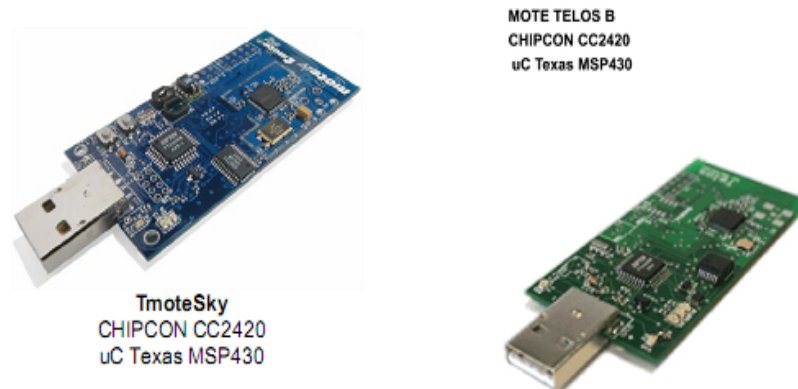


Figura 1: Motes Tmote y TelosB

TelosB y Tmote se refieren a plataformas hardware únicamente, El software que las acompaña se compone de:

Un sistema operativo diseñado para dispositivos con limitaciones en memoria, capacidad de procesamiento y consumo de potencia llamado TinyOS.

Un lenguaje de programación diseñado específicamente para TinyOS llamado NesC (Network Embedded System C), que es un dialecto de C basado en componentes.

Un Pequeño conjunto de herramientas en java para: escuchar un puerto serial con posibilidad de re direccionamiento de trafico a un puerto TCP, crear objetos mensaje compatibles con TinyOS, y ejemplos de programación de aplicaciones cliente.

Características de TelosB:

250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver.

Interoperabilidad con otros dispositivos IEEE 802.15.4.

Microcontrolador Texas Instruments MSP430 (8MHz frecuencia, 10k RAM, 48k Flash)

ADC, DAC, Integrado, Fuente de alimentación Supervisada, y controlador DMA.

Antena integrada en el PCB, 50m rango en interiores/ 125m rango en exteriores.

Consumo de potencia ultra bajo (1.8mA modo activo, 5.1µA modo inactivo).

Rápido cambio de modo inactivo a modo activo (<6µs).

Encriptación y autenticación en la capa de enlace de hardware.

Programación y recolección de datos vía USB.

16-pines de expansión y un conector opcional para antena SMA.

Características de TinyOS

Es un sistema operativo de código abierto diseñado para WSN escrito en NesC.

Los programas para TinyOS están escritos en NesC, y pueden ser configuraciones o componentes.

Los Componentes son los bloques operativos, son similares a un objeto, implementan y/o usan servicios para y de otros componentes, y tienen estado.

Los servicios relacionados son agrupados en interfaces, las interfaces son proporcionadas o usadas por los componentes.

Las Configuraciones “cablean” las interfaces declaradas por los componentes.

TinyOS está organizado en una arquitectura llamada HAA (hardware Abstraction Architecture) que consta de tres capas: HPL (H. Presentation Layer,) HAL (H. Abstraction Layer), HIL (H. Independent Layer).

El hardware de la plataforma TelosB es modelado por componentes de NesC y pertenece a la capa HPL.

2.2. Plataforma Sentilla

En agosto de 2007, fue anunciada la plataforma para computación ubicua Sentilla cuya característica principal es que ofrece nodos inalámbricos o motes con características del estado del arte actual programables directamente en java, esto representa un aumento inmenso en la cantidad de desarrolladores disponibles para impulsar el avance y masificación de uso de la computación ubicua, esto hizo que en la conferencia JavaOne en San Francisco USA en mayo del 2008, Sentilla fuera galardonada con el premio Duke's Choice a la innovación.

Como fue mencionado anteriormente, Moteiv adoptó el nombre Sentilla a partir del último trimestre de 2007. La empresa tomó la decisión de replantear la forma en que los motes se programaban, cambiando de TinyOS como sistema operativo y NesC como lenguaje de programación, a JavaPoint (una mini maquina virtual de java) como sistema operativo de los motes y Java CLDC como lenguaje de programación, además desarrollaron varias herramientas para facilitar el diseño tales como programación y monitoreo remoto de la red de sensores, y una IDE basada en eclipse entre otras.

Los cambios en la programación de motes resultaron en un cambio de paradigma pues se pasó de de una plataforma hardware y software libre, es decir total libertad de uso, a una plataforma hardware y software propietaria; con una curva

de aprendizaje fuerte, con herramientas más sencillas para el desarrollo pero dependiente de lo que la empresa ofrezca.

2.2.1. Modelo de Programación Sentilla.

La plataforma Sentilla facilita el desarrollo, despliegue y administración de sistemas y soluciones de Pervasive-Computing, mediante un conjunto de servicios para dispositivos (Sentilla Point), aplicaciones (Sentilla Work) y administración de la red (Sentilla HostServer). Esta plataforma conforma un modelo de programación distribuida, y proporciona un amplio conjunto de Java API's y servicios tanto para motes como para servidores.

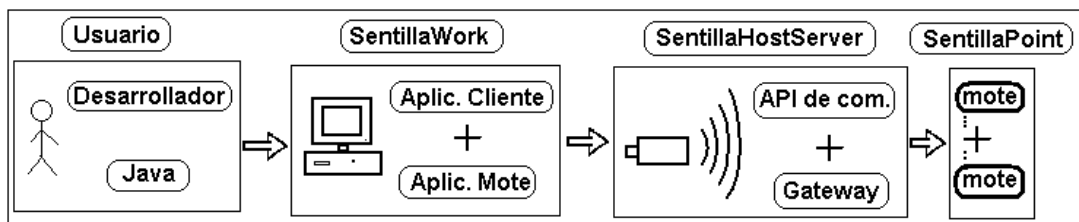


Figura 2: Diagrama de modelo de programación Sentilla

a. Sentilla Work

Es un entorno integrado de desarrollo de software basado en eclipse que incluye plug-ins y herramientas para facilitar al desarrollador el proceso de escritura, verificación, y despliegue de aplicaciones de Pervasive Computing. En Sentilla-Work se encuentran herramientas tales como: Editor de código y compilador, tutoriales, librerías de desarrollo para motes, librerías de desarrollo para clientes, aplicación registro e instalación de motes, estado de la red de motes, aplicaciones para el manejo de los motes, seguimiento de excepciones para el facilitar la corrección de aplicaciones tanto mote como cliente, aplicaciones de ejemplo tanto para motes como para servidor.

b. Sentilla Point

Viene instalado en los motes. Es un entorno de operación que comprende el Kernel Sentilla, servicios del sistema y librerías, una JVM con capacidades de red y capacidad para integrar dispositivos al mote y comunicarse con otros. Algunas características de Sentilla point son: Soporte para drivers y sensores, capacidad de Control y Administración de Aplicaciones, capacidad de comunicación, mensajería, networking, naming y addressing, seguridad, autorización y autenticación.

c. Sentilla HostServer

Es un servicio del sistema que proporciona conectividad entre la red Sentilla Point, aplicaciones cliente y Sentilla Work, mantiene y proporciona un estado e imagen consistente de la red Sentilla Point y el entorno de ejecución de las aplicaciones. Sentilla HostServer viene empaquetado e instalado con Sentilla Work por medio de la ejecución de Sentilla HostServer como un servicio, la red Sentilla Point puede ser constantemente monitoreada y administrada.

2.2.2. Características del hardware Sentilla

Las características hardware de los motes de esta empresa se mantienen bajo un hermetismo propio de las empresas que comercializan bajo licencia propietaria, solo mencionan que aún se maneja el procesador MSP430 de TI y el radio Chipcon CC2420 en los motes, además se disminuyó el tamaño del dispositivo con respecto a su predecesor TelosB, la figura 3 corresponde a un mote básico, que los desarrolladores de Sentilla adaptan como Gateway o como dispositivo final.

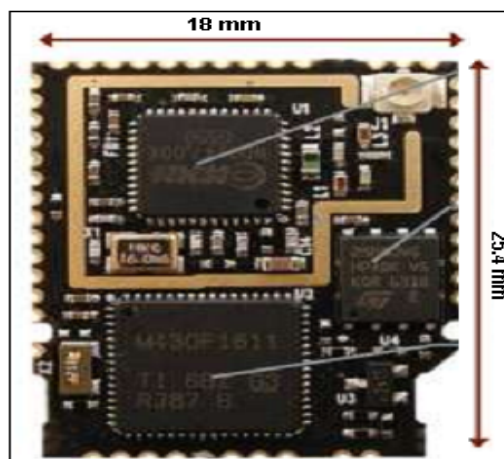


Figura 3: Chip básico de los motes Sentilla.

Para este proyecto se adquirió un kit PERK de evaluación que proporciona tanto hardware como software con limitaciones con respecto a la versión completa, y su hardware se describe a continuación:

a. Gateway USB

Es un mote básico adaptado como Gateway, re direcciona toda información que recibe en su puerto IEEE 802.15.4 hacia el puerto USB donde el Sentilla HostServer la recibe y la entrega a la aplicación de usuario que este conectada a él.

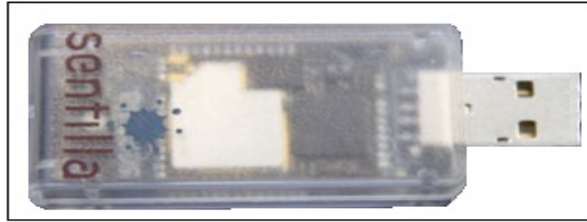


Figura 4: Gateway USB Sentilla

b. Motes JCreate

El kit PERK incluye dos motes JCreate, uno de ellos se muestra en la figura 5, son motes concebidos como dispositivos finales, cuentan con el chip básico, y traen conectores para periféricos los cuales se listan en la tabla 1:

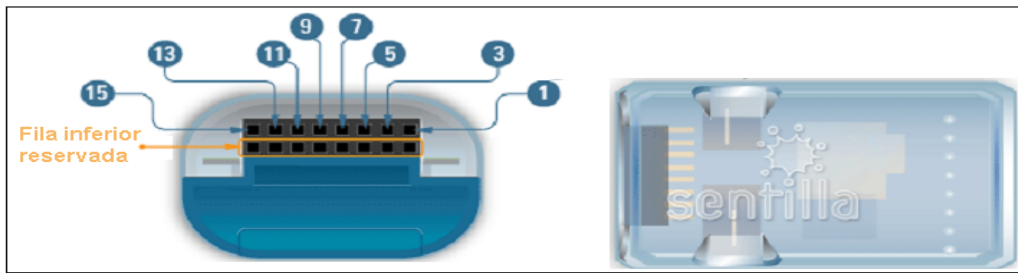


Figura 5: Mote JCreate Sentilla

# de Pin	USO
1	Tierra
3	Alimentación
5	ADC 2
7	ADC 3
9	GIO 0 (0x15)
11	GIO 1 (0x16)
13	Receptor UART
15	Transmisor UART

Tabla 1. Conectores para periféricos disponibles en los JCreate

c. Accesorio para instalación de firmware.

El firmware de los JCreate no se puede programar remotamente, para esta tarea el kit Perk incluye un conector USB que se adapte a los motes, para programar el firmware de los motes vía USB desde Sentilla Work, como se muestra en la figura 6.

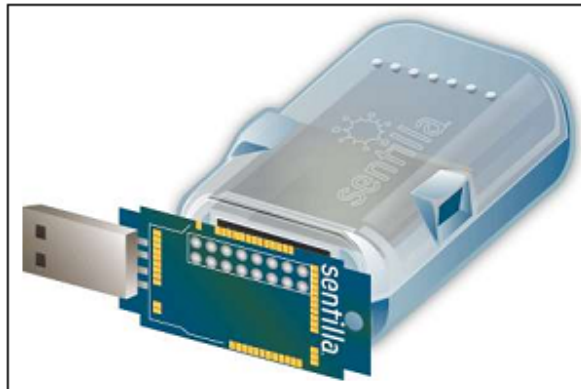


Figura 6: Accesorio para programación de firmware.

2.3. Plataforma XBee

La plataforma XBee es comercializada por MaxStream bajo licencia propietaria, tiene un enfoque a aplicaciones industriales que se evidencia en su programación mediante comandos AT por puerto serial, Sus principales características son:

- Tasa de transmisión de 250kbps, radio ISM 2.4GHz integrado en el PCB.
- Soporta redes beacon y non beacon (con coordinador) del protocolo 802.15.4
- Operación bajo el protocolo ZigBee
- ADC y conexión de líneas* digitales de entrada y salida entre dispositivos
- Consumo de 50mA en modo de espera, 10 μ A modo inactivo.
- Rápido cambio de modo inactivo a activo (<6 μ s)
- Encriptación y autenticación en la capa de enlace de hardware.
- Conectividad con cualquier dispositivo externos compatible con UART
- Programación y recolección de datos vía Serial

Cuenta con dos modos de operación para los motes, el modo transparente en el cual el mote transmite por radio la información que recibe por su puerto serial y viceversa, en este modo el mote actúa como sustituto de una línea serial; El segundo modo de operación es el modo API, en este modo la comunicación con el mote se realiza a través de una interface estructurada, la información es comunicada en tramas que contienen campos con propósito y orden definido, La API especifica como los comandos, respuestas a comandos y mensajes de estado del modulo son enviados y recibidos usando una trama de información

UART, la estructura general de una trama UART se muestra en la figura 7.

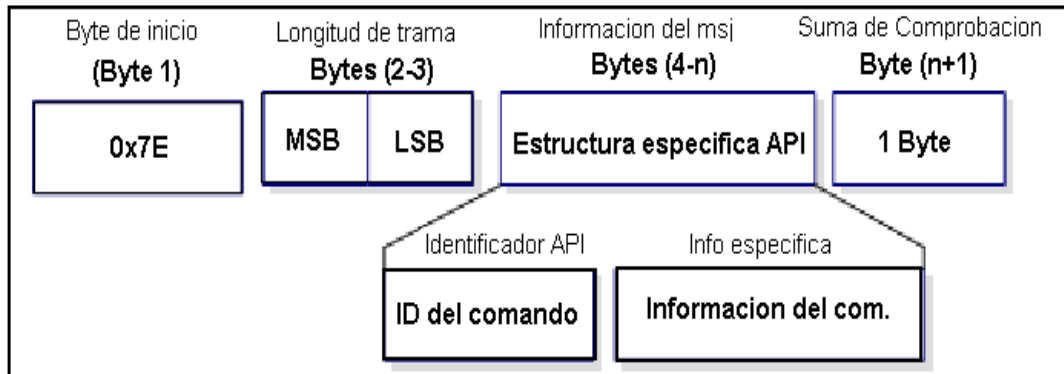


Figura 7: Trama UART del modo API

3. PRUEBAS REALIZADAS A LA PLATAFORMA SENTILLA

3.1. Alcance y potencia de transmisión de los motes en exteriores

El objetivo de esta prueba fue evaluar la potencia de transmisión y el alcance del radio de los motes, adicionalmente contrastar el dispositivo medidor de intensidad de señal recibida o por su sigla en ingles RSSI(Receive Signal Strength Indicator) de los motes contra el del un Sniffer aprovechando la bidireccionalidad de la prueba realizada.

Esta prueba se realizo en la pista de atletismo de la Universidad Industrial de Santander, la cual es al aire libre, sin edificaciones o muros que la rodean, para la prueba se utilizo un Sniffer 802.15.4 modelo CC2420, un pc-portátil con Sentilla Work y el software Packet Sniffer de Texas Instruments instalados, un Gateway USB Sentilla y cuatro motes JCreate con baterías nuevas; El PC corriendo Sentilla Work con el Gateway conectado a el sera llamado de ahora en adelante “base”.

Para esta prueba se creo un solo tipo de mensaje bautizado JustRssi que sera intercambiado entre la base y los motes, cuyo único campo es un numero tipo short, pensado para contener valores de RSSI.

Para los JCreate se creo e instalo una aplicación que dejaba a los motes a la espera de un mensaje JustRssi vacío proveniente de la base (PC + Gateway) al cual al ser recibido se le obtenía el valor de RSSI, este valor se almacenaba en el mismo mensaje recién recibido y se devolvía a la base. En la base se instalo una aplicación cliente que ejecutaba una función similar a un ping, enviaba mensajes JustRssi individuales y en orden secuencial a los motes (utilizando la

dirección MAC de cada uno) intentando hasta cuatro veces en caso que un mote no respondiera al mensaje de la base, antes de proseguir a hacer ping al siguiente mote.

Una vez un mensaje con el valor RSSI respectivo era recibido en la base proveniente de un mote, el Sniffer lo detectaba y obtenía el valor RSSI del mismo. Los dos valores es decir el RSSI del mensaje base-mote obtenido en el mote y enviado a la base, y el RSSI del mensaje mote-base recién recibido y obtenido por el Sniffer eran registrados.

El experimento se realizó dejando fijos el Sniffer y la base al inicio de la pista en un soporte de dos metros de altura, justo enfrente del soporte fijo se posicionaron cuatro motes JCreate en un soporte idéntico al primero cuya distancia al soporte fijo se varió en incrementos de 50 cm, iniciando desde 0 cm hasta perder comunicación con los motes encuestando los 4 motes y obteniendo los pares RSSI para cada uno en cada posición del soporte móvil.

En resumen se obtuvieron para cada mote dos vectores de valores RSSI, uno obtenido por el Sniffer y otro obtenido por los motes; Promediando los vectores provenientes del mismo instrumento se redujo la información RSSI de los cuatro motes a dos vectores: uno de valores tomados por el Sniffer y uno de valores tomados en los motes.

3.2. Alcance y potencia de transmisión de los motes en interiores

En el tercer nivel del edificio laboratorios pesados de la UIS se realizó una prueba para evaluar el impacto en el funcionamiento de los motes en un medio que tiene obstáculos tales como paredes y lugares cerrados. Se aplicó la misma metodología de la prueba en exteriores, con la diferencia que no se tomó la información correspondiente a una sola trayectoria recta, sino tres trayectorias o radios, con la base y el Sniffer como centro, y los motes alejándose gradualmente, un diagrama se muestra en la figura 8, para la trayectoria dos la pared presenta una perforación a tres metros de altura de aproximadamente 50 cm cuadrados.

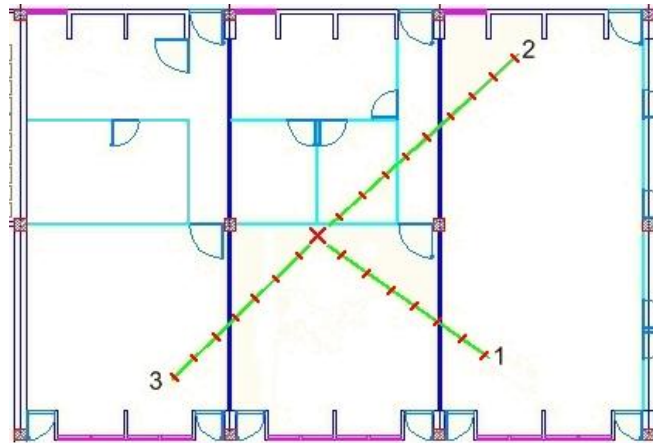


Figura 8: Plano del sitio y trayectorias de medida del experimento.

Igualmente los 8 vectores obtenidos para cada trayectoria fueron reducidos a dos, promediando los vectores de la misma fuente.

3.3. Prueba del protocolos multsalto de diseminación

En esta prueba se evaluó la efectividad del protocolo de diseminación proporcionado por Sentilla,, mediante el registro del tiempo que tardaba el mensaje en llegar a cada mote a partir de su envío desde la base; Para esta prueba se utilizó un Gateway USB conectado a un PC corriendo SentillaWork, y cuatro motes JCreate con baterías nuevas.

Para la correcta realización de la prueba se desarrollaron dos herramientas software: una para sincronizar los relojes de los motes con la base y otra para evaluar el rango de alcance del radio de un mote en un entorno cerrado arbitrario; La primera se utilizó para tener una base de tiempo común para registrar los tiempos de llegada de los mensajes a los motes y la segunda para disponer los motes en el laboratorio de manera que la cantidad de saltos de un mensaje durante su diseminación en la red fuera controlable.

El experimento se llevó a cabo con los motes dispuestos de las dos formas mostradas en la figura 9, las dos disposiciones fueron diseñadas para que un mensaje enviado desde una aplicación cliente utilizando el protocolo de diseminación fuera forzado a seguir la trayectoria M1-M2-M3-M4 para la disposición en línea, L de ahora en adelante, y la trayectoria M1-M2-(M3,M4) para la disposición en Y.

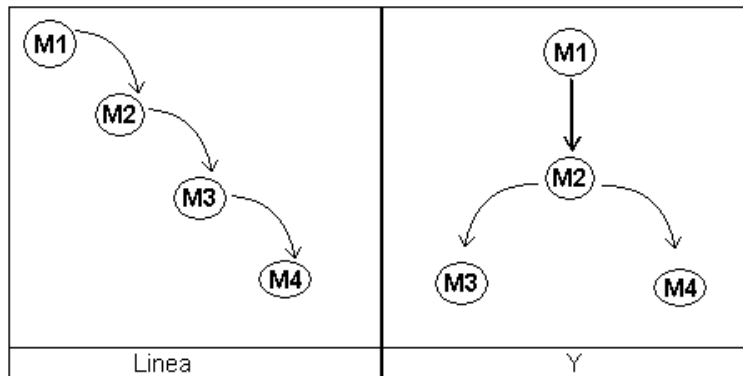


Figura 9: Disposición de los motes para las pruebas de diseminación

El primer paso en la aplicación de la prueba fue la sincronización de los relojes de los motes y la base, para esto todos los motes debían estar en el rango de la base; Luego se ubicaban los motes en la disposición escogida(L o Y) y a base comenzaba la emisión de 16 mensajes cada uno emitido solamente hasta que su predecesor ya hubiera sido difundido a toda la red; cada mote registraba en un vector el tiempo de llegada de cada mensaje; al terminar la emisión de todos los mensajes los motes eran llevados de vuelta al rango de alcance de la base donde sus vectores eran recogidos para posterior análisis.

Las pruebas fueron realizadas en el siguiente orden, primero se realizaron pruebas para determinar la carga útil de los mensajes, con el propósito de recoger la mayor cantidad de datos al final de la prueba, luego se aplicó la prueba con la disposición Y, y finalmente la disposición en línea. Se registró el nivel de batería antes y después de cada prueba.

3.4. Exploración de funcionalidades

Se experimentó con los dispositivos periféricos del JCreate con el objetivo de obtener conocimientos prácticos útiles para trabajos posteriores.

El JCreate cuenta con 4 convertidores analógicos digitales, dos puertos digitales de entrada o salida, y capacidad de comunicación UART.

También un acelerómetro Freescale MMA7260QT cuyo controlador es parte de la API para motes entregada por Sentilla. Se utilizó la clase AcceleratorButton que permite utilizar el acelerómetro como un botón cuya pulsación equivale a sacudir el mote.

4. RESULTADOS

4.1. Alcance y potencia de transmisión de los motes en exteriores

El alcance máximo fue 13.5m es decir 27 mensajes fueron recibidos por lo menos por uno de los cuatro motes, en promedio cada mote perdió 9 mensajes, En la figura 10 se muestran los valores RSSI tomados por los motes y los tomados por el Sniffer y en la tabla 2 se resumen los resultados.

Distancia máxima de comunicación entre base y mote	13.5m
% de mensajes perdidos por cada mote durante la prueba	30.00%

Tabla 2.: Resumen de resultados

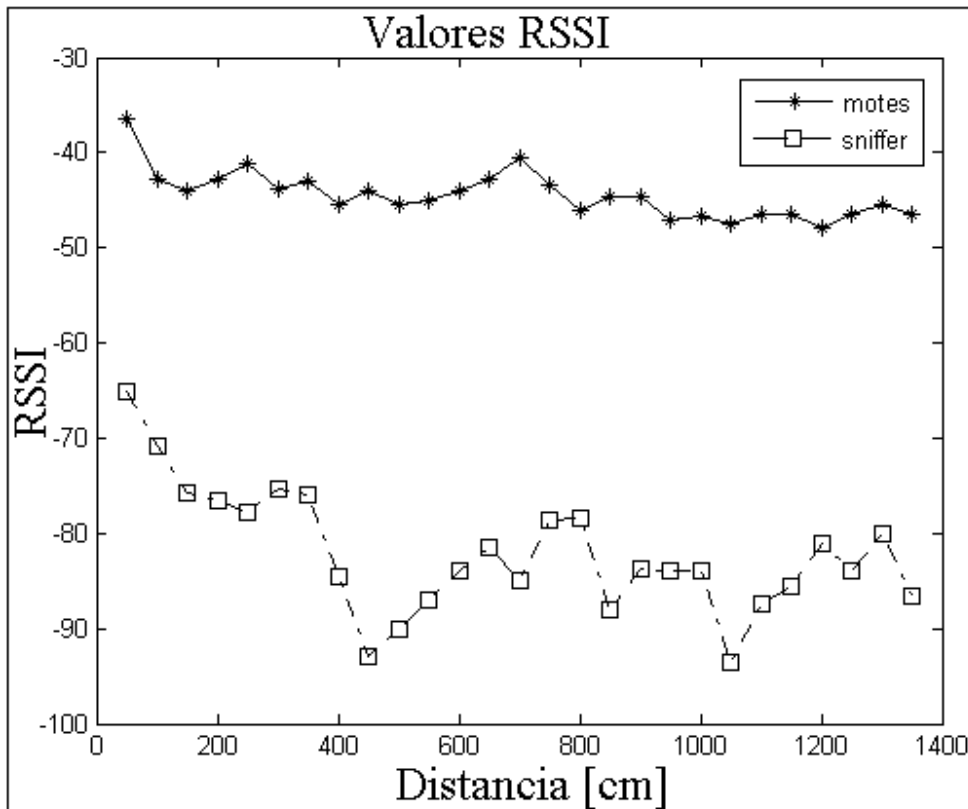


figura 10: valores RSSI tomados por los motes y por el sniffer

4.2. Alcance y potencia de transmisión de los motes en interiores

En interiores el alcance promedio fue de 7 metros, en la figura 11 se muestran los valores RSSI obtenidos para cada una de las tres trayectorias, junto con un diagrama explicativo con detalles relevantes acerca de la toma.

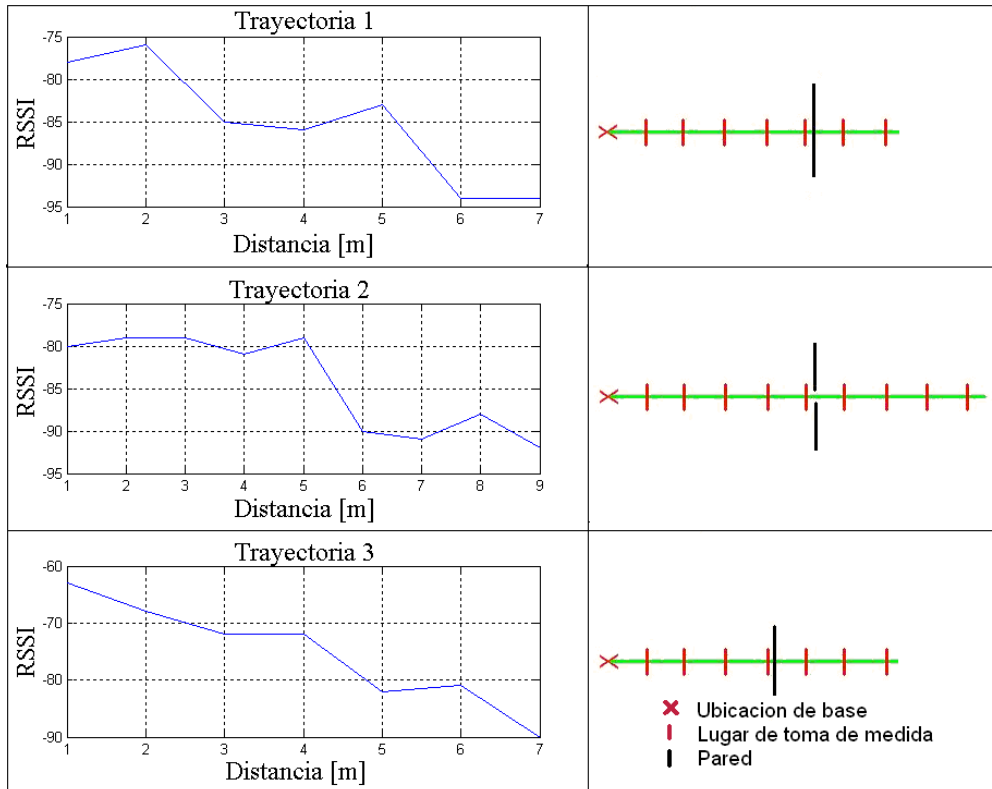


Figura 11: valores RSSI para cada trayectoria indoor.

4.3. Prueba del protocolos multisalto de diseminación

Para cada una de las disposiciones se grafica el tiempo total transcurrido desde el envío desde la base del mensaje hasta la llegada a cada mote y el tiempo que toma cada mensaje para realizar el salto de un mote a otro.

Durante la aplicación de esta prueba también se registro el consumo de baterías, Cada aplicación completa de la prueba consumió aproximadamente

un 10% de las baterías, la prueba a la disposición Y se realizo con las batería al 90% y la prueba a la disposición L al 80%.

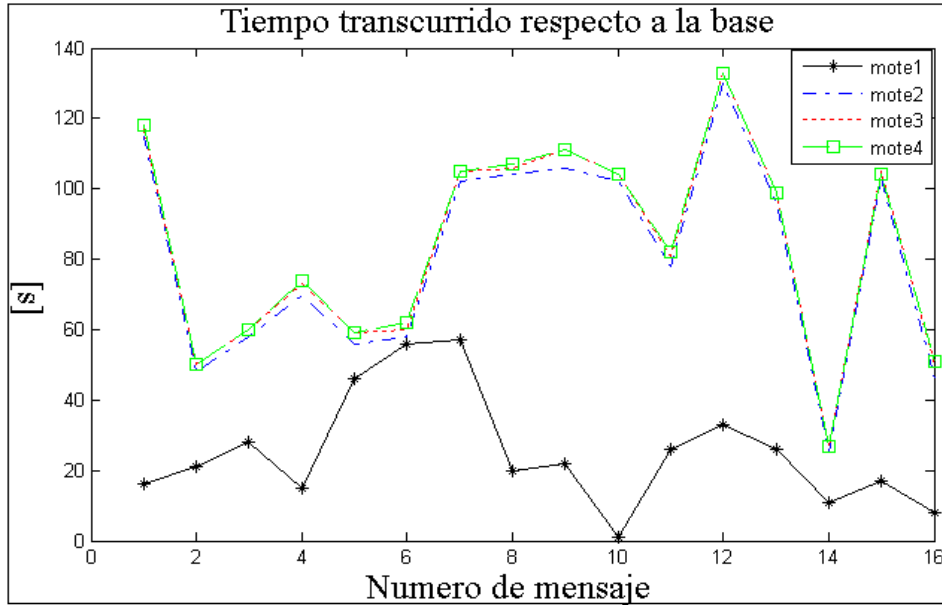


Figura 12: Tiempos totales de transito en la disposición Y

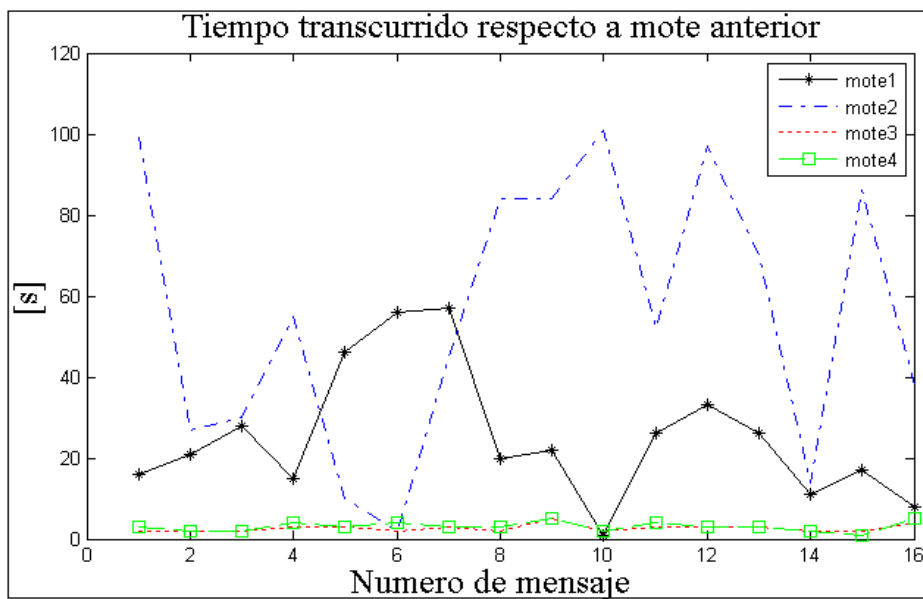


Figura 13: Tiempos de salto relativos para la disposición Y.

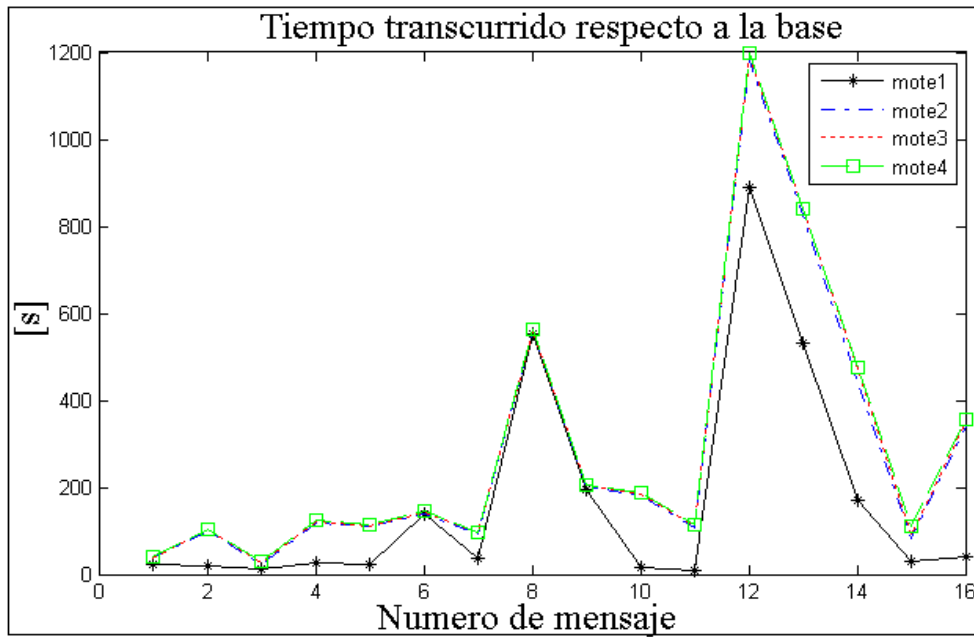


Figura 14: Tiempos totales de transito en la disposición L

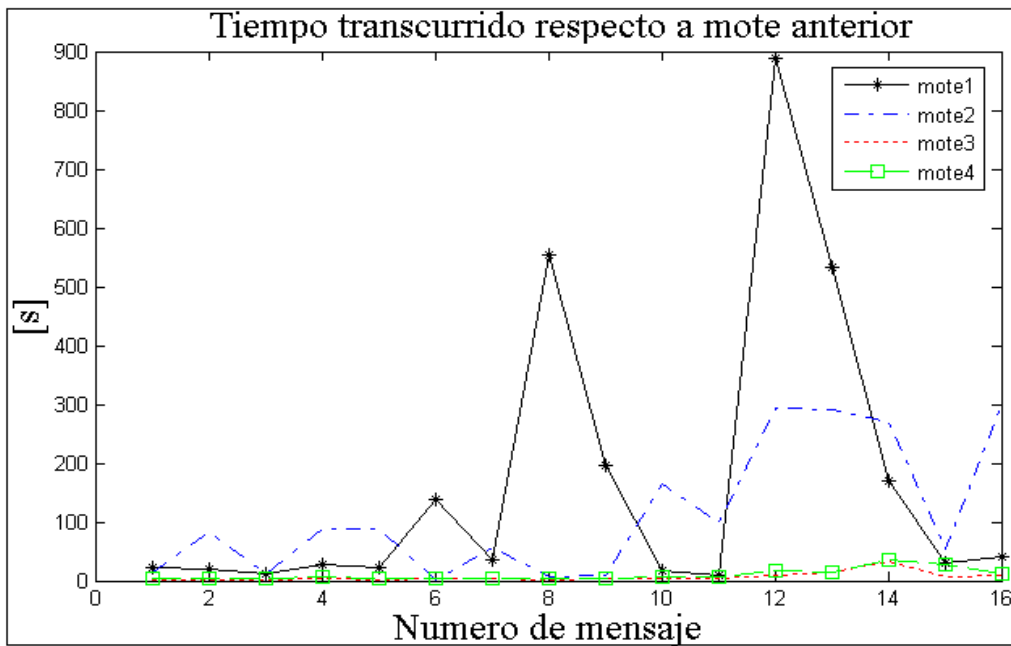


Figura 15: Tiempos de salto relativos para la disposición L.

Mote1	25[s]
Mote2	56[s]
Mote3	2 [s]
Mote4	3 [s]

Tabla 3.: promedios relativos de transito para la disposición Y

Mote1	169[s]
Mote2	114[s]
Mote3	6 [s]
Mote4	10 [s]

Tabla 4.: promedios relativos de transito para la disposición L

4.4. Exploración de funcionalidades

Todas las interfaces de la tabla 1 fueron evaluadas y el código correspondiente se encuentra en el apéndice 2.

5. ANALISIS Y DISCUSION

5.1. Alcance y potencia de transmisión de los motes en exteriores

Al observar la figura 10, es evidente el desfase de los valores RSSI medidos por los motes y los medidos por el Sniffer esto es debido a que los valores RSSI que provee el registro del radio necesitan un ajuste empírico, hallando la diferencia entre los valores del Sniffer y del mote se propone un ajuste a los valores tomados por el mote de -37, en la figura 16 se muestran los valores RSSI del mote ajustados junto con los valores del Sniffer.

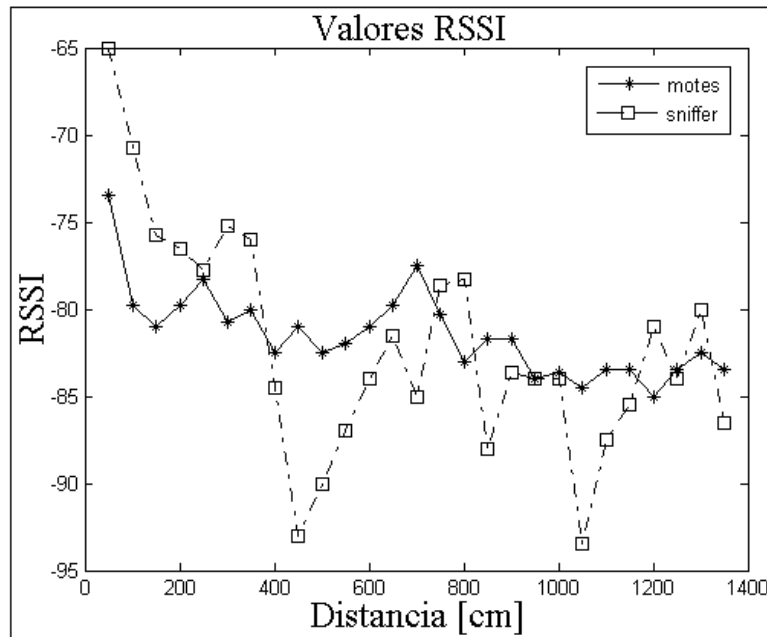


Figura 16: Valores RSSI tomados por el Sniffer junto con los del mote corregidos.

Para contrastar los datos contra un modelo teórico, se tomaron los datos del Sniffer como base para los cálculos puesto que este es lo más cercano que se tiene a un instrumento patrón para medidas de RSSI;

Dado que los datos recogidos por el Sniffer corresponden a la potencia de los mensajes entrantes recibidos en la base, estos valores deben cumplir con la siguiente ecuación:

$$Pr = Pt + (Gtx + Gtr) - (Ltx + Lrx) - LOS$$

Pr = potencia recibida [dB]

Pt = Potencia transmitida [dB]

Gtx = Ganancia en la antena transmisora [dBi]

Gtr = ganancia en la antena receptora [dBi]

Ltx = pérdida en la línea de transmisión [dB]

Ltr = pérdida en la línea de recepción [dB]

LOS = Pérdidas en el aire; especificadas mediante la ecuación de Friss

Ahora:

Según la ecuación de Friss

$$\underline{LOS=32.44+20*\log(F)+20*\log(D)+Kf}$$

F=frecuencia de transmisión [MHz]

Kf=constante de ajuste [dB]

D=distancia de transmisión [Km]

Entonces despreciando las perdidas en las líneas, la ecuación que describe teóricamente los datos es:

$$\underline{Pr=(Pt+Gtx+Grx+Kf)-20*\log(F)-20*\log(D)-32.44}$$

se toma (Pt+Gtx+Grx-Kf) como una constante K, y se despeja para obtener un valor aproximado utilizando los datos.

Como los valores RSSI están en dBm, se les resta 30 para convertirlos a dB, antes de utilizarlos.

la ecuación para obtener valores de K usando los datos experimentales queda:

$$\underline{K= Pexp+20*\log(2445)+20*\log(r/1000)+2.44 [dB]}$$

Pexp=datos RSSI tomados;

El valor promedio obtenido para K evaluando los valores RSSI fue: -25.0149 [dB].

Entonces

$$\underline{Pt=k-(Gtx+Grx)}$$

Suponiendo que las ganancias de recepción y transmisión de las antenas del Sniffer y el mote sumadas son mayores que 5 dB, se puede presumir que el mote está transmitiendo a valores por debajo de 0dBm, lo que cae dentro del rango de valores posibles, tomando como referencia los motes TelosB.

la gráfica teórica de el valor de potencia recibido, se muestra junto con los datos experimentales en la Figura 17

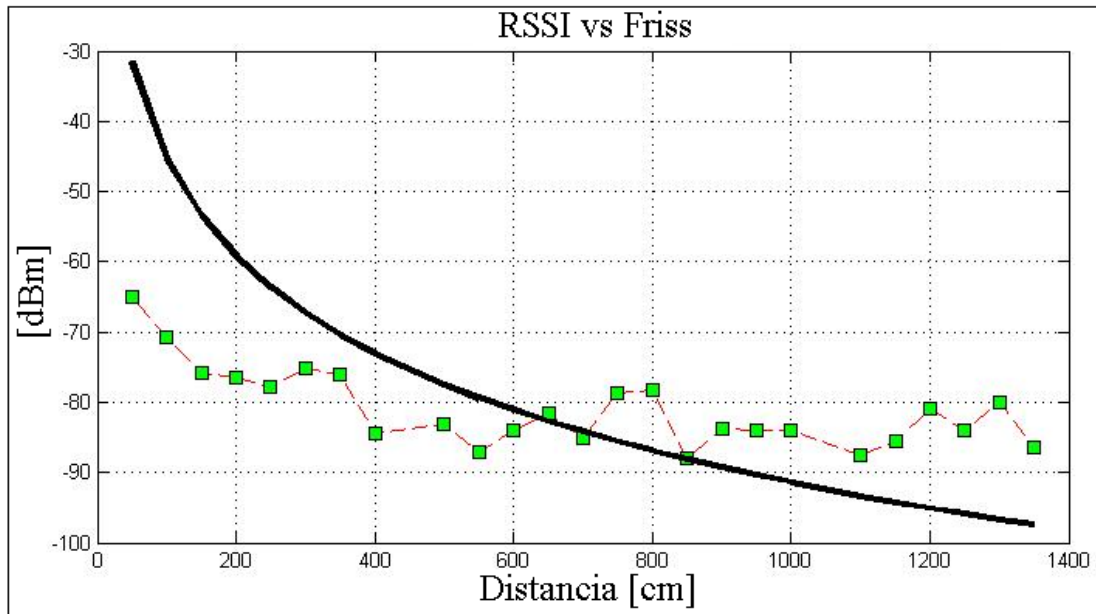


Figura 17: Gráficas teórica y practica para potencia recibida

En resumen se encontró:

los motes JCreate tienen un alcance muy pobre que representa una limitación seria a la hora de utilizarlos en aplicaciones reales, además su tasa de perdida de mensajes debe ser tomada en cuenta en el caso de utilizarlos para la adquisición de información sensible, en este caso se recomienda implementar un mensaje de confirmación de recepción (ACK).

La ausencia de un ACK y posibilidad de reenvío de mensajes para los cuales no reciba, se debe posiblemente al hecho que el radio es el dispositivo que mas energía consume, y la implementación de un ACK duplicaría el trafico de la red, disminuyendo la vida de las baterías a la mitad.

5.2. Alcance y potencia de transmisión de los motes en interiores

El alcance en interiores es muy corto, Esto limita el rango de aplicaciones en interiores de la plataforma, por ejemplo un mote en una habitación solo se puede comunicar con motes que estén en habitaciones contiguas.

En la figura 11 se puede observar que la atenuación de la señal debido a una pared es de alrededor de 10dB, además se observó que se obtuvo un alcance adicional de dos metros en la trayectoria dos en la cual había un hueco en la pared.

5.3. Prueba del protocolo multisalto de diseminación

Para la prueba al protocolo de diseminación se escogió el tamaño del vector de tiempos que se recolectó en la base al final de la prueba, basados en la carga útil disponible en los mensajes de radio; Según el protocolo 802.15.4 esta es de 87 bytes, pero hay que restar lo que utilizan los desarrolladores de Sentilla para implementar las características de la plataforma, tales como encriptación del canal de radio, monitoreo del estado de la red, y protocolos multisalto.

Se determinó por prueba y error que el tamaño de la carga útil después de las implementaciones de Sentilla es de 64 bytes aproximadamente.

Al observar el desarrollo de la prueba desde el inspector de tráfico de SentillaWork, se notó que los mensajes enviados con el protocolo de diseminación no son enviados inmediatamente después de ejecutado el comando enviar en el programa cliente, sino periódicamente (aproximadamente cada minuto) junto con los mensajes de control y monitoreo que envía SentillaWork los cuales están abstraídos al usuario.

Estos mensajes de control pueden resultar inconvenientes a la hora de diseñar aplicaciones de bajo consumo de potencia, pues no es posible detenerlos ni controlar su periodo de circulación por la red.

En las gráficas 13 y 15 los tiempos de salto en ambas disposiciones son notablemente mayores para los primeros dos motes; Es posible que la forma en que esta implementado el protocolo recargue la responsabilidad de los cálculos en los motes mas cercanos a la base por ser la via de paso obligada de los mensajes de control del resto de la red; esta conclusión tentativa se refuerza al ver que con la disposición Y el segundo mote se demora mas tiempo en recibir los mensajes que en la disposición L posiblemente al ser responsable del trafico y tareas computacionales para los motes tres y cuatro.

Ademas En la gráfica 15 se ve claramente en el mote 1 y un poco menos pronunciada en el mote 2 una tendencia de aumento en los tiempos de salto a medida que aumentan los mensajes diseminados ; esto puede ser debido a que el salto adicional de esta disposición hace que aumente aun mas la carga computacional para los motes cercanos y que las tareas que deben realizar se acumulen, esto combinado con la periodicidad de la circulación de los mensajes de diseminación hace que el tiempo que tardan en recibir los mensajes aumente considerablemente.

5.4. Exploración de funcionalidades

La forma en que Sentilla maneja los dispositivos resulta muy conveniente, pues brinda interfaces sobre las cuales es posible construir drivers.

Por ejemplo existe una interfaz para los ADC del JCreate, que permite conectar en ellos cualquier dispositivo y configurarlo rápidamente, El driver para el acelerometro esta utiliza esta interfaz.

Para los pines digitales existe la interfaz DigitalPin; para los 8 leds integrados en el JCreate se utiliza esta interfaz como base para crear un driver para los leds que los abstrae y los presenta al usuario como un dispositivo de alto nivel con funciones de apagar y prender todos los leds al tiempo aplicando una mascara de bits, esto resulta muy útil para identificar errores en las aplicaciones mote.

En los casos de los acelerómetros y los pines digitales, si se desea aumentar la velocidad de ejecución se recomienda acceder los métodos de la interfaces directamente en vez de los drivers de alto nivel.

La interfaz UART también realiza funciones de alto nivel pero se hace necesaria la utilización de un circuito para empalmar la lógica CMOS de mote ($V_{dd}=3v$) a la lógica TTL que se utilizan los dispositivos con puerto RS232.

6. AGRADECIMIENTOS

A mis maestros.

REFERENCIAS

- [1] Rúgeles U. José de Jesús, Redes de sensores inalámbricos: Diseño implementación y caracterización de un prototipo de red ZigBee en ambientes abiertos y cerrados. Tesis de Magister, Escuela de ingenierías eléctrica y electrónica, UIS, Santander, Colombia (2007).
- [2] Crosbow; TelosB; datasheet
- [3] Moteiv; Tmote Sky; Datasheet (11/13/2006)
- [4] TinyOs community; TinyOS tutorials
- [5] Maxstream; Product manual XBee OEM RF Modules 802.15.4.

- [6] IEEE; Std 802.15.4TM-2006; estandard
- [7] Sentilla; Sentilla Works user Guide; Manual
- [8] Sentilla; Sentilla Programmer's guide; Manual
- [9] Chipcon; manual CC2420