

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Máster Oficial en Sistemas Inteligentes y Aplicaciones Numéricas en
Ingeniería



TRABAJO FIN DE MÁSTER
**Comunicación robusta punto a punto con módulos embebidos de radio
frecuencia XBee**

Rubén Valentín Torón Domínguez

Tutores: Jorge Cabrera Gámez
Antonio Carlos Domínguez Brito

Diciembre 2013

Índice General

Resumen.....	4
1. Introducción	6
1.1. Programación Remota u OTAP (Over The Air Programming)	12
1.2. Mula de Datos (Data mule)	14
1.3. Origen y justificación del trabajo	15
1.4. Objetivos	16
2. Estado del Arte.....	17
2.1. Redes sensoriales o Wireless Sensor Networks (WSN).....	17
2.2. Programación remota (OTAP).....	18
2.3. Mula de datos (Data mule).....	19
3. Hardware.....	20
3.1. Datos generales de waspmote v1.1 de Libelium.....	21
3.2. Consumo en los diferentes modos.....	21
3.3. Entradas/salidas	21
3.4. Sensores en la placa	22
3.5. XBee Pro RF 868	22
4. Implementación	23
4.1. Trama XBee	23
4.2. Trama OTAP de Libelium	24
4.3. Protocolo robusto	25
4.4. Mejoras en la API de Libelium.....	29
4.5. Shell OTAP Linux.....	30
4.6. Programación remota y mula de datos en waspmotes	31
5. Metodología	32
6. Resultados	38
6.1. Shell OTAP	38
6.2. OTAP entre waspmotes.....	44
6.3. Shell OTAP primera versión.....	47
7. Conclusiones	48
7.1. Líneas futuras	49

Referencias.....	50
Índice de figuras	53
Índice de tablas	55
Apéndice 1.....	56
A.1. Shell OTAP - Programa A.....	56
A.1. Shell OTAP - Programa B.....	66
A.1. OTAP entre waspmotes - Programa A	76
A.3. Potencia de la señal - Programa A	85

Resumen

Una de las capacidades más demandadas en redes de sensores inalámbricos (WSN) es la de poder reprogramar los nodos sensores de manera remota a través de la infraestructura de comunicaciones existente. Con frecuencia esta infraestructura está basada en pequeños módulos de radiocomunicación sometidos a limitaciones de ancho de banda, tamaño de paquete de datos y de ciclo de trabajo (duty cycle). Esta circunstancia, unida a las limitaciones computacionales de los nodos sensores, y a que la transmisión de nuevos programas a los nodos implica la transmisión de ciertos volúmenes de datos hace que mantener una comunicación sostenida en el tiempo con la red de sensores se convierta en un proceso complejo y frágil.

En este Trabajo de Fin de Máster se aborda el desarrollo de un protocolo de comunicaciones robusto punto a punto que gestiona los paquetes, permite el restablecimiento de la comunicación en el punto donde se produce el error, sincroniza el emisor y el receptor, y controla el flujo de datos para evitar la saturación de los waspmotes.

Este protocolo se ha ensayado experimentalmente sobre las motas de la empresa Libelium denominadas waspmotes, dispositivos de bajo consumo diseñados para formar redes inalámbricas de sensores. Las comunicaciones se han resuelto con módulos embebidos de radio frecuencia XBee-PRO 868 instalados en los waspmotes y en los equipos que sirve de punto de acceso a la red sensorial. Estos módulos permiten una comunicación entre dispositivos limitada a paquetes de tamaño reducido.

Se ha conseguido demostrar el funcionamiento de este protocolo de comunicaciones en el desarrollo de aplicaciones de programación remota (OTAP) y de transporte de datos (data muling) sobre diferentes distancias de enlace. Asimismo, se demuestra la superioridad de las aplicaciones OTAP basadas en este protocolo frente a otras soluciones basadas en protocolos no coordinados.

Palabras clave: redes de sensores inalámbricos (WSN), mota, waspmote, mula de datos (data muling), programación remota (OTAP), XBee-PRO, control de flujo

1. Introducción

Las redes sensoriales o Wireless Sensor Networks (WSN) [1] permiten monitorizar nuestro entorno. Están formadas por una serie de pequeños dispositivos electrónicos que captan datos del exterior a través de sensores, procesan los datos y los intercambian de manera inalámbrica. Estos dispositivos son conocidos como motas (mota de polvo, traducción de la palabra inglesa mote) por su pequeño tamaño y porque pueden estar situados en cualquier lugar.

La empresa española Libelium [2] ha desarrollado su propia mota, el waspmote. Este dispositivo, inspirado en la plataforma de hardware libre Arduino [3], mantiene una filosofía modular, para conectar y utilizar en cada mota multitud de placas de sensores, y añadir diferentes sistemas de comunicación que permitan crear una red sensorial.



Figura 1: Waspote: mota de Libelium

Los waspmotes están orientados a cualquier tipo de uso doméstico o empresarial. La versión 1.1 del waspmote de Libelium, está compuesta por un micro controlador Atmel ATmega1281 de 8 bits a una frecuencia de 8MHz, una memoria Flash de 128Kb para albergar el programa a ejecutar, una memoria SRAM de 8Kb para las instrucciones y variables que están siendo ejecutadas y

1. Introducción

una memoria EEPROM de 4Kb para las variables que se mantienen aún reiniciando el waspmote.

Un waspmote puede integrar hasta ocho sensores, entre un catálogo de más de cincuenta disponibles, entre los que se encuentran sensores de radiación, humedad, CO₂, etc.

Para crear redes sensoriales, los waspmotes pueden integrar diferentes módulos de comunicación, desde conexión a redes telefónicas móviles por GPRS, módulos Wi-Fi o módulos Bluetooth, hasta módulos por radiofrecuencia.

Entre los módulos de radiofrecuencia que se pueden integrar, destacan los módulos XBee [4] de Digi [5], que utilizan diferentes protocolos de comunicación en frecuencias libres de licencia. Con sus módulos XBee-PRO 868 [6] se obtiene una comunicación de hasta 12 kilómetros [7] de separación entre waspmotes con antenas unidireccionales de 5dBi de ganancia.

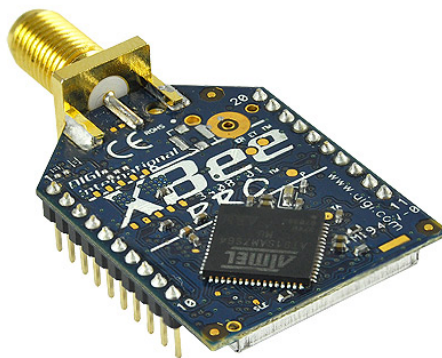


Figura 2: módulo XBee-PRO 868 de Digi

Los módulos XBee-PRO 868 operan en la frecuencia de 868MHz, el ancho de banda que utilizan es de 24Kbps, y las comunicaciones pueden ir encriptadas. Funcionan a 3.3V, consumen 500mA para transmitir a la máxima potencia y 65mA para recibir un paquete. Disponen de cinco niveles de potencia de transmisión: 1mW, 25mW, 100mW, 200mW y 300mW.

1. Introducción



Figura 3: Adaptadores USB para XBee [8]

Para configurar los módulos XBee se utiliza un adaptador USB para XBee y el programa X-CTU [9] distribuido por el fabricante Digi. Con el programa X-CTU podemos configurar numerosos parámetros del módulo XBee (p.e. para cambiar la potencia que utilizará el módulo para transmitir y recibir, el nombre o identificador del módulo, etc.), y podemos también leer información de campos de sólo lectura, como el contador del número de errores de paquetes recibidos, número de errores de paquetes enviados, etc.

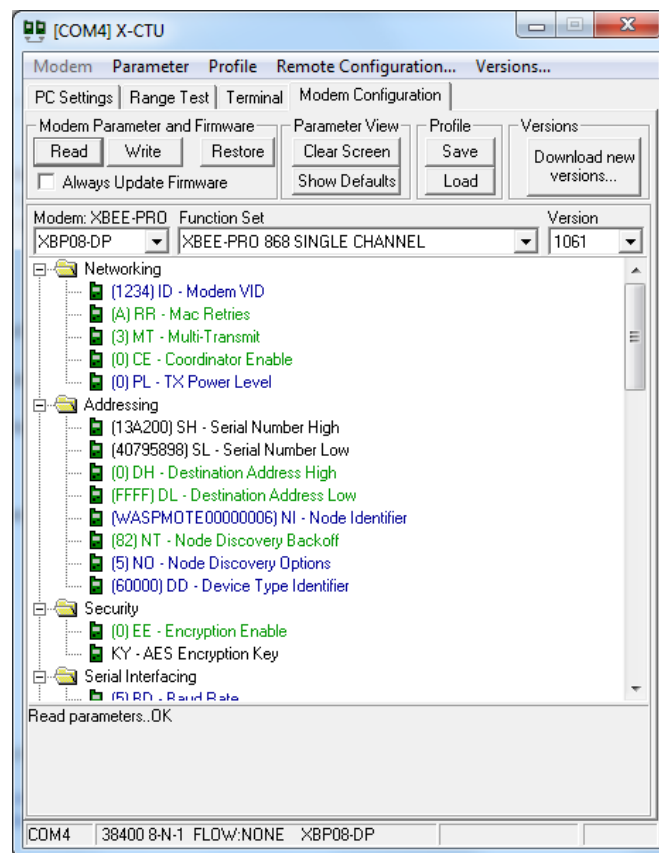


Figura 4: Programa X-CTU de Digi para programar los módulos XBee

1. Introducción

Los módulos XBee tienen dos modos de comunicación [10] a través del puerto UART al que se conectan en la placa del waspmote o al adaptador USB: modo transparente, donde lo que recibe por el puerto UART lo transmite directamente por radiofrecuencia, y lo que recibe por radiofrecuencia lo transmite al puerto UART; y modo API, donde lo que se envía y recibe está contenido en tramas que pueden activar operaciones o eventos en el módulo. El modo API tiene dos submodos: modo API 1 sin caracteres de escape de control, para enviar datos binarios; y modo API 2 con caracteres de escape de control, para enviar datos con ciertos caracteres de escape.

En modo transparente es necesario utilizar comandos AT para modificar o leer los parámetros del módulo XBee. Los comandos AT pueden tener como objetivo el módulo XBee del waspmote local, o ser enviados a otro waspmote remoto para que sea ejecutado remotamente.

En modo API se especifican cómo tienen que ser las tramas para enviar comandos AT. A diferencia con lo que ocurre con el modo transparente, los cambios de parámetros no se aplican inmediatamente, sino que en modo API los cambios son puestos en una cola y serán aplicados cuando se envíe una trama que lo indique.

Los comandos AT pueden tener como objetivo el módulo XBee del waspmote local, o ser enviados a otro waspmote remoto para que sea ejecutado remotamente.

Existen tres modos de transmisión [11], dependiendo del número de módulos receptores del mensaje: si es un solo receptor, el modo es unidifusión (unicast); si existe más de un receptor, el modo es multidifusión (multicast); y si los receptores son todos los que existen en la red, el modo es difusión (broadcast). En modo difusión y multidifusión, el waspmote emisor transmite a varios waspmotes receptores de manera simultánea de una sola vez.

1. Introducción



Figura 5: Modo de transmisión unidifusión (unicast)



Figura 6: Modo de transmisión multidifusión (multicast)



Figura 7: Modo de transmisión difusión (broadcast)

En una red sensorial con módulos XBee-PRO 868, cuando un waspmote quiere enviar un mensaje a otro waspmote, debe realizar los siguientes pasos:

- Activar módulo XBee-PRO 868.
- Preparar el contenido mensaje (payload).

1. Introducción

- Configurar la cabecera del mensaje, indicando dirección del waspmote destino, tipo de mensaje, tamaño del mensaje, etc.
- Configurar la cola del mensaje, indicando el bit de control (checksum), etc.
- Almacenar el mensaje completo, paquete, en el buffer de envío del módulo XBee-PRO 868.
- Utilizar la API (Application Programming Interface) para ordenar el envío del paquete.

Cuando el módulo XBee-PRO 868 recibe la orden de envío, se realizan los siguientes pasos:

- Envía el paquete al módulo XBee-PRO 868 del waspmote destino.
- Si el modo de transmisión es unicast y el módulo XBee-PRO 868 emisor y destino están configurados para intercambiar mensajes de confirmación de recepción (ACK), el módulo XBee-PRO 868 del waspmote destino envía un mensaje de confirmación de recepción al módulo XBee-PRO 868 del waspmote emisor.
- Si el modo de transmisión es unicast, y no se recibe el mensaje de confirmación en menos de 200ms, y el módulo XBee-PRO se ha configurado con un número de reintentos mayor a cero, se reenvía el mensaje.

Todo este proceso se realiza de manera transparente para los waspmotes, a nivel hardware de módulos XBee-PRO 868, la única condición es que ambos módulos estén activos.

Existen varios factores que podrían impedir la correcta comunicación entre dos waspmotes:

- Interferencias o ruido en el canal de transmisión, que provoquen una alteración del contenido del paquete, el cambio de un bit, podría afectar a la dirección de destino, o al tipo del paquete o al contenido del mensaje.
- Distancia entre módulos XBee-PRO 868 y nivel de potencia empleada. Una potencia inadecuada podría afectar a la comunicación, ya sea, por

1. Introducción

transmitir con escasa potencia para la distancia de enlace que se desea cubrir, donde los paquetes no llegarían; o por transmitir con excesiva potencia para distancias de enlace muy cortas, produciendo intermodulación [12]. Este último problema solo se suele presentar cuando las radios se encuentran a pocos metros de distancia.

- Pérdida de sincronización entre los módulos XBee-PRO 868, si la confirmación del mensaje no llega, el emisor lo reenvía, pero puede ocurrir que la confirmación del receptor se haya enviado y llegue después de 200ms.

Cuando el módulo XBee-PRO 868 del waspmote receptor recibe el mensaje, lo almacena en un buffer de recepción del módulo XBee-PRO 868 a la espera de que el waspmote receptor acceda a este buffer para leer su contenido y lo almacene en un vector de mensajes pendientes de tratar en la memoria del waspmote.

1.1. Programación Remota u OTAP (Over The Air Programming)

El concepto de OTAP o Programación Inalámbrica [9], comenzó a popularizarse con la reprogramación de dispositivos móviles, como Smartphones. En nuestros días, es muy común actualizar el firmware de nuestro Smartphone sin conectarlo a nuestro PC mediante un cable USB, simplemente a través de una conexión a una red Wi-Fi o 3G.

En 2007, Libelium y Arduino, desarrollaron una placa que ampliaba las posibilidades de comunicación de Arduino, junto con la familia de productos XBee, ofreciendo la posibilidad de usar bandas de frecuencias libres de licencia, como son la 2.4GHz, 900MHz y 868MHz, con protocolos abiertos como 802.15.4, ZigBee y Digimesh o protocolos propietarios del RF 900 y RF 868.

Dependiendo del protocolo, podríamos alcanzar los 12 km [7] de separación entre nodos usando antenas unidireccionales de 5dBi, siempre en condiciones ideales. Esto nos permite montar una red de sensores inalámbricos muy distantes entre sí sin necesidad de añadir el coste adicional de una licencia por uso de frecuencia.

1. Introducción

La presencia de los módulos de radio, además de facilitar la comunicación de las motas, permite realizar OTAP desde cualquier punto de la red. Asimismo, es posible modificar la programación de una mota ejecutando un nuevo programa almacenado en la tarjeta microSD o descargando una nueva versión de un programa, sin necesidad de acceder físicamente al waspmote. Por ejemplo, si el waspmote está en una boya no tendríamos que coger una embarcación para acercarnos, o si está instalado en una caja estanca, no tendríamos que desmontarla.

Libelium incluye una implementación de OTAP en su API, en la que se emplea una aplicación para PC en línea de comandos para realizar el OTAP con los waspmote. El OTAP de Libelium soporta módulos XBee que operan con diferentes protocolos, 802.15.4, ZibBee, Digimesh, RF 900 y RF 868, y también funciona con GPRS y con Bluetooth.

Entre las ventajas del OTAP implementado por Libelium se incluyen:

- Actualización o cambio de firmware de manera inalámbrica.
- Recuperar un waspmote si no responde.
- Descubrir nodos con un mensaje de difusión o broadcast.
- Dependiendo del tipo de módulo XBee, la posibilidad de seleccionar un canal de radio para desarrollar OTAP con el waspmote destino que no interfiera con el resto de waspmotes.

Los pasos para realizar OTAP comprenden:

- Localizar el waspmote a actualizar.
- Comprobar versión actual del firmware.
- Enviar nuevo firmware.
- Reiniciar el waspmote.
- Recuperar waspmote si el proceso falla en algún punto.

1. Introducción

1.2. Mula de Datos (Data mule)

El concepto de Transporte o Mula de Datos, proviene de Data MULE (Mobile Ubiquitous LAN Extension) [13], se aplica al dispositivo que permite intercomunicar nodos de una red que normalmente no son accesibles. Un posible ejemplo de Transporte o Mula de Datos lo tenemos cuando un vehículo o mula se aproxima a un nodo aislado, carga los datos almacenados en ese nodo, lo reprograma y cuando conecta de nuevo con una red de destino descarga la información con destino esa red.

La operación de un servicio postal nos proporciona un magnífico ejemplo de transporte de datos. Cuando queremos comunicarnos con otra persona, dejaremos la carta en un buzón de envío, cuando la mula de datos, en este caso el cartero, tenga prevista en su ruta la recogida y entrega de cartas, pasará por el buzón a recoger lo que queremos enviar y depositará en nuestro buzón personal las cartas que van dirigidas a nosotros.

Este tipo de comunicaciones entre el emisor y el receptor se realiza de forma asíncrona y no en tiempo real, como se aprecia en el ejemplo del servicio postal. El medio para la comunicación es la mula de datos.

En nuestro caso, tendremos waspmotes localizados en ubicaciones muy remotas e incomunicados entre sí, y con ayuda de un vehículo, que transportará otro waspmote, irá acercándolo a cada uno de los waspmotes, para que entre en cobertura y comience la recolección de los datos que hayan recogido los sensores desde la última conexión y la recepción de los paquetes que tengan como destino el waspmote al que se ha acercado. Una vez finalizada la recogida y la recepción de datos, el vehículo continuará su ruta al siguiente waspmote, así hasta recorrerlos todos.

La descarga de los datos recolectados puede realizarse al alcanzar un destino concreto o al entrar en contacto con la infraestructura de red que permitirá inyectar esos datos en la red y que lleguen a destino.

1.3. Origen y justificación del trabajo

Existen escenarios donde la pérdida de un paquete, ya sea por no recibirlo o por no tratarlo a tiempo, puede ser crítica debido a que la comunicación implique el intercambio de un elevado volumen de datos, como ocurre con la programación remota o el transporte de datos (data muling). En ambos casos la pérdida de un paquete puede derivar en un reinicio del proceso y que deban retransmitirse de nuevo todos los paquetes.

El módulo XBee-PRO 868 está limitado por el ciclo de carga máximo (duty cycle) de un 10%; en otras palabras, en una hora se alcanza el ciclo de carga máxima después de seis minutos de emisión continua de paquetes. Este ciclo de carga máxima está fijado por dos motivos. En primer lugar, el estándar ETSI EN 300 220-1 [14] fija el ciclo de carga máxima en 10% para garantizar el uso compartido del canal disponible. Por otra parte, la transmisión continua de paquetes, por intercambio elevado de volumen de datos o por retransmisión por pérdida de paquetes, provoca que el módulo XBee-PRO 868 se sobrecaliente, y por protección y para disminuir la temperatura del módulo, el módulo deja de transmitir. El waspmote estaría incapacitado para transmitir hasta que se restablezca el módulo XBee-PRO 868.

Se evidencia, por tanto, la necesidad de mejorar la comunicación para lograr una comunicación robusta entre dos waspmotes y resolver de manera eficiente las limitaciones hardware y los problemas de intercambio de paquetes.

Un protocolo de intercambio de paquetes indexados, contribuiría a continuar el proceso desde el punto en que se interrumpió el proceso, evitando el reinicio continuado en cada interrupción, y su relación directa con el sobrecalentamiento del módulo.

1.4. Objetivos

El objetivo principal de este Trabajo Fin de Máster es el desarrollo de un protocolo de comunicación robusto que gestione el intercambio de un volumen elevado de paquetes entre dos waspmotes. Este protocolo será la base para la programación remota y otros servicios de transporte de datos.

La programación remota nos ayudará a conseguir una mayor autonomía y prestaciones en una red de sensores. Proporcionará mecanismos para poder actualizar de manera inalámbrica el programa que se ejecute en el waspmote o cambiarlo por otro programa totalmente diferente. Supondrá, además, un elemento de comodidad para el programador que use waspmotes de versiones anteriores a la 1.2. En la versión 1.1 es necesaria la desconexión física del módulo XBee cada vez que hay que reprogramar el waspmote. Gracias al OTAP es posible reprogramar esta versión del hardware sin tener que desconectar la radio en un proceso donde primero se compila el programa para generar el fichero hexcode y después se descarga en el waspmote usando el shell OTAP.

Es importante destacar la reprogramación de los nodos de la red puede realizarse de dos formas diferentes, bien transmitiendo un nuevo programa mediante OTAP, bien indicando al nodo que debe ejecutar un nuevo programa de entre los almacenados en su tarjeta microSD. Esta posibilidad permite obviar de manera elegante y eficiente la limitación hardware de 128KB de programa principal, y disponer del espacio disponible en la memoria SD como tamaño máximo de conjunto de programas. Asimismo, mediante OTAP es posible actualizar uno de los programas almacenados en la tarjeta sin que el resto se vea afectado.

En nuestro contexto, la red de sensores estará constituida por waspmotes fijos y móviles, donde existirá un waspmote que actuará como mula de datos e irá siguiendo una ruta para acercarse a cada uno de los waspmotes aislados de la red, realizar la descarga de los datos almacenados en esos nodos en la mula de carga, y eventualmente reprogramarlos, bien enviando un nuevo programa de manera inalámbrica o una solicitud de ejecución de uno de los programas almacenados en la tarjeta SD del waspmote remoto.

2. Estado del Arte

2.1. Redes sensoriales o Wireless Sensor Networks (WSN)

Las redes sensoriales es una tecnología emergente que está siendo empleada en un número creciente de aplicaciones, y que puede considerarse como la tecnología subyacente en el concepto de Internet de las cosas o Internet of Things (IoT) [15], definido como red de objetos cotidianos interconectados.

Actualmente existe la demanda en el mercado de conectarlo todo a internet, y tenerlo todo controlado, desde cuándo ha terminado la lavadora hasta saber a qué hora apagamos la luz de la entrada. Las redes sensoriales no se quedan al margen de esta tendencia, formarán parte del Internet de las cosas, donde la red de sensores aportará información importante al resto de nodos del Internet de las cosas.

Las redes sensoriales no se limitan al aire como medio físico de comunicación, engloba cualquier medio de transmisión que no sea el cable, como por ejemplo el agua [16], la luz, aunque son las tecnologías de comunicación basadas en señales ópticas, básicamente en el infrarrojo, y las que emplean señales de radio frecuencia (Bluetooth, WI-FI, radio en bandas ISM libres de frecuencia, ZigBee, GSM, 3G,...) las más utilizadas.

Los últimos estudios de redes sensoriales quieren solucionar los grandes problemas de las redes inalámbricas:

- Baja eficiencia [17], no se utiliza el camino óptimo de saltos para comunicar a dos nodos, o no se utiliza una recuperación de errores óptima.
- Alto consumo energético, un mal ajuste de la potencia hace que se gaste más energía de la necesaria.
- Protocolo para nodos móviles [18], donde la topología de la red varía y tenga en cuenta el ciclo de carga máxima de cada nodo.

2. Estado del arte

En este trabajo nos centraremos en las conexiones punto a punto, estudiaremos e implementaremos un protocolo que aumente la eficiencia respecto a la recuperación de errores.

2.2. Programación remota (OTAP)

Los móviles llevan años utilizando este concepto de programación remota. La posibilidad de actualizar nuestro terminal sin conectarlo a un ordenador, simplemente utilizando una conexión Wi-Fi o la conexión con nuestro operador de telefonía móvil.

La naturaleza de las redes sensoriales hace necesario la utilización de este concepto ya que, tras el despliegue inicial de los sensores, las tareas de mantenimiento y actualización del software de la mota son más convenientes si se desarrollan de manera remota [19]. No siempre tenemos acceso directo a una mota después de su despliegue, como una mota sumergida en el agua o instalada en lo alto de una estructura.

Las líneas activas de trabajo de este ámbito son las que se centran en:

- Implementar la programación remota a micro controladores que no disponen de él [20].
- La recuperación ante fallos de programación remota, cuando un programa no funciona como estaba previsto e inutiliza la mota [19].
- Implementar la programación remota en redes a gran escala [21], como ciudades que tengan monitorizados todos los espacios de aparcamiento de la calle.

El waspmote de Libelium ya implementa la programación remota, en este trabajo se estudia y optimiza el código de Libelium para conseguir una programación remota robusta. En este sentido, el esquema de lazo abierto actual usado por Libelium impide que el emisor y el receptor se sincronicen, de manera que se balancee la carga de envío de paquetes, un problema de estabilidad que posee la API de Libelium y se soluciona con nuestro protocolo robusto. En general, este soporte es frágil como puede comprobarse a partir de los comentarios de los usuarios en el foro de Libelium [22].

2. Estado del arte

2.3. Mula de datos (Data mule)

Es un término poco conocido, pero no por ello menos importante. Algunas redes sensoriales crecen de tal forma, que algunas motas se encuentran aisladas y es muy costoso crear una comunicación directa con otro punto de la red.

Los últimos estudios intentan resolver los siguientes problemas de mula de datos:

- Ampliar la escalabilidad con múltiples mulas [23], a medida que crecen las motas, una sola mula es insuficiente, se deben añadir más, pero deben estar coordinadas.
- Generalización del problema del viajero [24], un problema NP-completo, computacionalmente muy costoso, en este contexto.
- Gestionar encuentro no programados, en redes donde los nodos son móviles, la mula se puede encontrar con un nodo [25] de forma no prevista.

En este Trabajo de Fin de Máster abordaremos la comunicación de ciertos volúmenes de datos de mota a mota, tanto para realizar un transporte de datos como para reprogramar remotamente una de las motas.

3. Hardware

El controlador hardware está basado en la versión 1.1 de waspmote de Libelium. El waspmote de Libelium, tiene numerosos elementos en común con las tarjetas Arduino como son el compilador y el entorno de desarrollo. Además en general, la mayoría de las librerías desarrolladas para Arduino son fácilmente reutilizadas sobre un waspmote.

No obstante, el waspmote integra un reloj de tiempo real (RTC, real-time clock), un acelerómetro de tres ejes, lector de tarjeta micro SD y acepta una variedad de dispositivos de comunicación sin tener que usar "shields" o tarjetas de interfaz de Arduino. Además, en el waspmote se pueden apagar selectivamente dispositivos, desconectar el regulador 5V-3.3V y dispone de algunos modos de bajo consumo o hibernación que no existen sobre Arduino.

Actualmente Arduino permite entrar en modo "sleep", modo de bajo consumo similar al waspmote, aunque no funciona igual y no utiliza un reloj de tiempo real (RTC) para generar una interrupción y despertar de nuevo al microcontrolador. Este reloj de tiempo real permite ser una mayor flexibilidad a la hora de determinar el momento en que se quiere despertar al waspmote.

Como corresponde a una mota, el waspmote está diseñado para cumplir con su función ahorrando al máximo el consumo de la batería. Para ello es fundamental que el consumo medio sea el menor posible y que la mota pase la mayor parte del tiempo apagada o "dormida" en un modo de bajo consumo.

3. Hardware

3.1. Datos generales de waspmote v1.1 de Libelium

- Microcontrolador: ATmega1281
- Frecuencia: 8MHz
- SRAM: 8Kb
- EEPROM: 4Kb
- FLASH: 128Kb
- Capacidad tarjeta SD: 2GB
- Peso: 20gr
- Dimensiones: 73.5 x 51 x 13mm
- Temperatura: [-10°C, +65°C]
- Reloj: RTC (32KHz)

Libelium indica en las especificaciones del waspmote v1.1, que la capacidad máxima de la tarjeta SD es de 2 GB, aunque según nuestras pruebas con tarjetas de 4 GB, el waspmote admite tarjetas de mayor capacidad a la indicada por el fabricante mediante un pequeño cambio en la configuración del software.

3.2. Consumo en los diferentes modos

- ON: 15mA
- Sleep: 55µA
- Deep Sleep: 55µA
- Hibernate: 0.07µA

3.3. Entradas/salidas

- 7 analógicas (I)
- 8 digitales (I/O)
- 1 PWM
- 2 puertos UART
- 1 bus I2C
- 1 USB
- 1 SPI

3. Hardware

3.4. Sensores en la placa

- Temperatura
- Acelerómetro en los tres ejes
- Nivel de batería

3.5. XBee Pro RF 868

El módulo XBee-Pro 868 ha sido el que se ha utilizado para nuestros experimentos.

Estos módulos se caracterizan por

- Utiliza la frecuencia libre de 868MHz.
- Ancho de banda de 24Kbps.
- Comunicación encriptada.
- Hasta 80km de separación entre módulos, según fabricante Digi, aunque Libelium lo reduce a 12km.
- 10% de ciclo de carga máximo para evitar dañar el hardware antes de alcanzar los 60°C.

En la siguiente tabla comparativa, podemos observar las diferencias con otros módulos de XBee de Digi.

Modelo	Protocolo	Frecuencia	Potencia	Sensibilidad	Distancia
XBee-802.15.4	802.15.4	2.4 GHz	1mW	-92dBm	500m
XBee-802.15.4-Pro	802.15.4	2.4 GHz	100mW	-100dBm	7km
XBee-ZB	ZigBee-Pro	2.4 GHz	2mW	-96dBm	500m
XBee-ZB-Pro	ZigBee-Pro	2.4 GHz	50mW	-102dBm	7km
XBee-868	RF	868MHz	315mW	-112dBm	12km
XBee-900	RF	900MHz	50mW	-100dBm	10km

Tabla 1: Comparativa módulos XBee de Digi

4. Implementación

4.1. Trama XBee

Todas las tramas XBee [10] se forman por tres bytes cabecera, una parte variable de carga útil y un byte final de checksum, como se muestra en la tabla 2.

Cabecera	Tamaño de la trama	Carga útil (payload)	Checksum
1 byte	2 bytes	variable	1 byte

Tabla 2: Estructura general de una trama XBee de Digi

Debe tenerse en cuenta que la estructura general de una trama es la siguiente:

- El primer byte, es el byte cabecera, que delimita e identifica el comienzo de la trama. Su valor siempre es 0x7D.
- Dos bytes para indicar el tamaño completo de la trama, actualmente no se transmiten tramas de más de 255 bytes, así que el primer byte será siempre 0x00, y el segundo byte indicará un valor de 0 a 255.
- La carga útil o payload, es variable.
- El último byte de la trama, es el byte de checksum, que se obtiene sumando todos los bytes de la carga útil y restando el resultado a 0xFF. Cuando un paquete se recibe, se comprueba que la suma del número de bytes de la carga útil más el byte del checksum es 0xFF. Los resultados de estos cálculos se limitan con el módulo 0x100.

En nuestro caso, más específico, se utilizará la estructura de trama en modo API, como se muestra en la tabla 3.

Cabecera	Tamaño del paquete	Tipo trama	Identificador trama	Dirección destino 64 bits	Reservado	Radio difusión	Opciones	Datos	Checksum
1 byte	2 bytes	1 byte	1 byte	8 bytes	2 bytes	1 byte	1 byte	variable	1 byte

Tabla 3: Estructura de trama API para el envío de datos a otro módulo

4. Implementación

Donde:

- Los tres primeros bytes cabecera, y el byte final de checksum, funcionan como se ha indicado para la estructura general de una trama XBee.
- Un byte para indicar el tipo de trama, en nuestro caso tendrá un valor a 0x10 si enviamos o 0x90 si recibimos el paquete.
- Un byte para identificar la trama, será un número correlativo a una secuencia de confirmación de trama.
- Los siguientes 8 bytes, indicarán la dirección fuente o destino en formato de 64 bits. La dirección broadcast es 0xFFFFFFFF.
- Dos bytes reservados, están siempre a 0xFFFE
- Un byte de radio de difusión, para indicar el número máximo de saltos en una transmisión de difusión.
- Un byte de opciones, puede indicar si está desactivado el envío de tramas de confirmación de recepción (ACK).
- Los bytes de datos son variables. El tamaño máximo son 100 bytes.

Cuando queremos enviar una trama, el tipo lo fijamos a 0x10 e indicamos la dirección destino en formato 64 bits. Cuando la trama es recibida correctamente, el receptor recibe la misma trama, pero el tipo es sustituido por 0x90, y la dirección que recibe es la de la fuente en formato de 64 bits. Estos cambios se realizan en el módulo XBee de manera transparente para el usuario.

4.2. Trama OTAP de Libelium

Libelium, para diferenciar una trama con datos con intercambio de información con una trama de programación remota, ha añadido unos bytes en el campo de datos.

0x3C	0x3D	0x3E	Tipo OTAP	Datos OTAP
1 byte	1 byte	1 byte	1 byte	variable

Tabla 4: Estructura del campo datos de un paquete OTAP de Libelium

4. Implementación

Los tres primeros bytes cabecera del campo de datos, son fijos a 0x3C, 0x3D y 0x3E. El byte tipo OTAP, indica el tipo de trama OTAP.

Si la trama que se envía es una trama que contiene un programa, se añade un nuevo campo, índice, que indica el número de trama que se está enviando y controla el orden en que se recibe.

0x3C	0x3D	0x3E	Tipo OTAP	Índice	Datos OTAP
1 byte	1 byte	1 byte	1 byte	1 byte	variable

Tabla 5: Estructura del campo datos de una trama OTAP para el envío de un programa

4.3. Protocolo robusto

Los waspmotes se han diseñado para minimizar el consumo de batería y no es improbable que estén en estado de hibernación, es decir, apagados, durante periodos prolongados. Aunque disponen de una pequeña memoria EEPROM, se hace indispensable el uso de tarjetas microSD como medio de almacenamiento masivo de datos. En general, el coste de las operaciones de lectura (escritura sobre este medio serán lentas y el waspmote permanecerá bloqueado mientras no se completen.

Los paquetes de datos que se reciben a través de la interfaz de radio de un waspmote son almacenados en un buffer circular de mensajes pendientes de tratar. Este buffer tiene un límite y, si se reciben mensajes pero no se tratan, los mensajes más antiguos se irán sobrescribiendo con los nuevos mensajes. Por tanto, es vital que durante las comunicaciones que involucren un elevado número de mensajes, el receptor se coordine con el emisor para ajustar la tasa de transmisión a la capacidad de procesamiento del receptor y no perder mensajes por desbordamiento del buffer de recepción.

En otras palabras, es necesaria una confirmación de disponibilidad que controle el flujo de tramas y no sature el buffer de mensajes pendientes de tratar, y detecte otros errores inesperados.

4. Implementación

Existen varios modelos de protocolos para controlar los errores en flujos de comunicación orientados a paquetes, entre ellos destacan [26]:

- Parada y espera
- Vuelta atrás N
- Repetición selectiva

Parada y espera es un protocolo para manejar errores en la trama transmitida. Cuando se transmite una trama, el emisor se queda a la espera, el receptor, recibe la trama, y si es correcta, envía una confirmación al emisor para que siga con la siguiente trama; o, en caso de error, envía una trama para indicar que debe volver a retransmitir la trama.

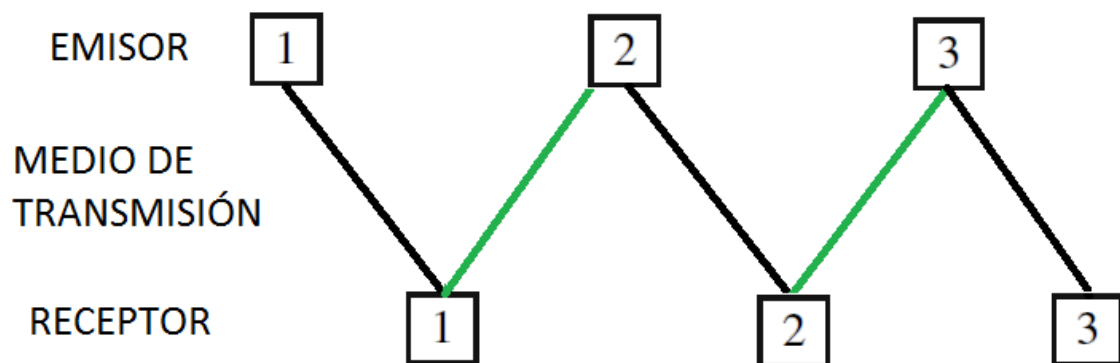


Figura 8: Protocolo parada y espera

Vuelta atrás N, es un protocolo que mantiene una copia en un buffer, denominado ventana de transmisión, las tramas que son enviadas. El emisor va transmitiendo tramas hasta completar la ventana de transmisión y, a medida que va recibiendo tramas de confirmación, va eliminando las tramas confirmadas de la ventana de transmisión, siempre y cuando las tramas anteriores a la trama cuya recepción se ha confirmado ya estén confirmadas. Si no recibe confirmación de una trama en un tiempo determinado o se recibe una trama indicando que se ha detectado un error, el emisor transmite todas las tramas de la ventana a partir de la última confirmada. Cuando la ventana de transmisión ha sido completamente confirmada, se añaden nuevas tramas a transmitir. Por ejemplo, si enviamos cuatro tramas, y recibimos confirmación de la primera y de la tercera y la cuarta, si falla la segunda, se reenvía la segunda, la tercera y la cuarta.

4. Implementación

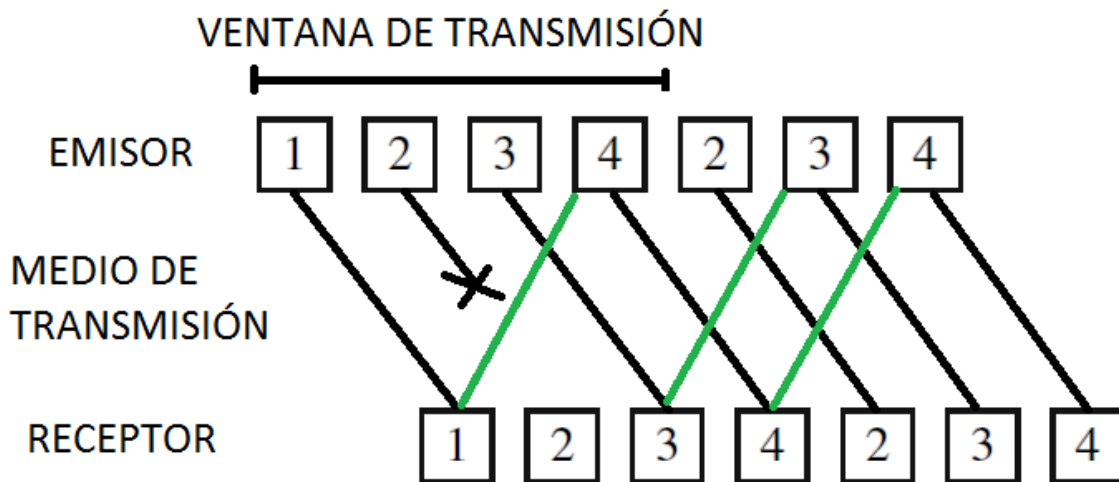


Figura 9: Protocolo vuelta atrás N

Repetición selectiva, es un protocolo que mantiene una copia en un buffer las tramas que son enviadas, este buffer se denomina ventana de transmisión, al igual que en vuelta atrás N. El emisor va transmitiendo tramas hasta completar la ventana de transmisión, a medida que va recibiendo tramas de confirmación, va eliminando las tramas confirmadas de la ventana de transmisión, y a diferencia de vuelta atrás N, no espera a que se transmitan todas las tramas de la ventana de transmisión, va añadiendo nuevas tramas. Si no recibe confirmación de una trama en un tiempo determinado, la reenvía.

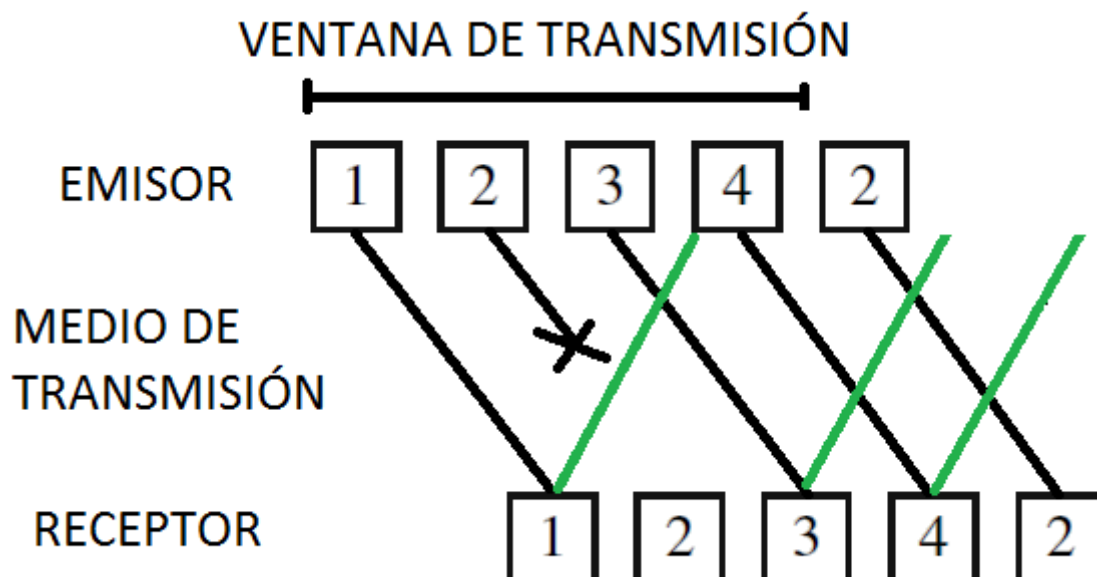


Figura 10: Protocolo repetición selectiva

4. Implementación

En este trabajo se ha implementado el protocolo de parada y espera sobre una versión modificada de la API de Libelium. En este protocolo el waspmote emisor envía una trama al waspmote receptor. El waspmote receptor recibe y procesa la trama recibida y cuando finaliza, envía una confirmación al emisor para que siga transmitiendo. Es una posición conservadora, pero nos asegura que se reciben y procesan cada uno de los paquetes de datos. En caso de error en la trama, el waspmote receptor avisa al emisor con una trama de error para que retransmita la trama errónea. Si en un periodo de tiempo, no se recibe confirmación, se retransmite.

Con este objetivo se han creado nuevos tipos de trama OTAP para las diversas confirmaciones que pueden existir en programación remota y en mula de datos.

Tipo de trama OTAP	Código
ACK_NEW_FIRM_END_FRAME	0x74
ACK_NEW_FIRM_RECEIVED_FRAME	0x75
ACK_NEW_FIRMWARE_MESSAGE_OK	0x76
ACK_NEW_FIRMWARE_MESSAGE_ERROR	0x77

Tabla 6: Códigos confirmación de trama OTAP

En algunas situaciones, sobre todo a largas distancias, si el módulo XBee está configurado para que envíe confirmación de recepción a nivel de módulo y reenvío en caso de fallo, puede ser que se envíe el paquete y que no llegue la confirmación en 200ms. En ese caso el waspmote emisor retransmite una trama que ha sido recibida correctamente aunque no en el tiempo esperado.

La confirmación de recepción a nivel de módulo es configurable: activada o desactivada. En el caso de que esté activada, se debe fijar un valor para el número de retransmisiones en caso de que no llegue la confirmación en 200ms. Si está desactivada, no existirán retransmisiones en caso de que no llegue la confirmación a nivel de módulo. El protocolo robusto implementa a nivel de software sus propias confirmaciones de tramas, independientemente de cómo estén configurados los módulos.

4. Implementación

El protocolo está diseñado para que en el lado del emisor se ejecute el siguiente algoritmo:

- Después de transmitir, esperar.
- Si se recibe confirmación, continuar con la siguiente trama.
- Si no se recibe confirmación, retransmitir trama y volver a esperar.
- Si se recibe una confirmación repetida de una trama anterior, no hacer nada, seguir esperando, ya que hemos enviado una nueva trama.

En el lado del receptor, el protocolo se articula de la siguiente forma:

- Después de recibir una trama, si no es correcta, enviar trama de error.
- Después de recibir una trama, si es correcta, procesarla.
- Después de procesar la trama, enviar confirmación al emisor.
- Si se recibe una trama ya procesada, reenviar confirmación de esa trama, puede ser que exista una desincronización, y en el lado del emisor se haya acabado el tiempo de espera, y por eso haya llegado la trama.

El servicio de transporte o mula de datos se basa en el mismo protocolo robusto, añadiéndose dos nuevos tipos de trama:

Tipo de trama mula de datos	Código
NEW_DATA_MULE	0x72
ACK_NEW_DATA_MULE	0x73

Tabla 7: Códigos de trama y de confirmación de trama para mula de datos

4.4. Mejoras en la API de Libelium

La última versión de la API de Libelium para waspmote v1.1 es la v0.32, pero Libelium ha diseñado y está comercializando el waspmote PRO (v1.2), una versión actualizada y mejorada de su waspmote v1.1. Con la salida al mercado de este nuevo waspmote, Libelium liberó una nueva versión de su API, no compatible con la versión anterior de waspmote. En la nueva API, Libelium ha eliminado funciones obsoletas, y ha limpiado y reducido el tamaño del API.

4. Implementación

En este trabajo se ha migrado el módulo de XBee dentro del nuevo API de Libelium, en principio disponible solo para la versión 1.2 del hardware, para que funcione también sobre la versión 1.1 del waspmote.

En este proceso se han introducido importantes mejoras en el módulo XBeeCore que una incidencia muy positiva en estabilidad, menor consumo de RAM y un menor tamaño del ejecutable. Un aspecto relevante en estas mejoras es que se ha rediseñado completamente el buffer de mensajes pendientes de tratar para garantizar que los mensajes se entregan en el orden en el que se han recibido, algo que en el API original no estaba garantizado.

4.5. Shell OTAP Linux

Para desarrollar el shell de OTAP en Linux, se ha utilizado la librería LibXBee [26]. Esta librería de C/C++ nos permite comunicarnos con los módulos XBee a través de un adaptador conectado al puerto USB, enviando y recibiendo mensajes de otros módulos XBee. En este proyecto se ha usado la última versión disponible (v3.09) de esta librería.

Este shell nos permite:

- Consultar los programas almacenados en la tarjeta SD del waspmote.
- Cargar un programa remotamente.
- Eliminar un programa de la tarjeta SD.
- Ejecutar uno de los programas almacenados en la tarjeta SD.



Figura 11: Shell OTAP Libelium

Para implementar nuestro shell, no hemos tenido acceso al código fuente del shell de Libelium, pero sí teníamos el código del API del waspmote.

4. Implementación

El primer paso fue modificar el API para que mostrara todos los mensajes que recibía el waspmote del shell de Libelium, y así implementar nuestro shell que funcionara igual y sustituyera al shell de Libelium.

El segundo paso fue implementar el shell en C++ en Linux con la librería LibXBee v3.09. Se realizaron pruebas y se comprobó que el waspmote, tras un determinado número de paquetes recibidos, el buffer de paquetes pendientes de procesar, se llenaba al máximo, y al llegar un nuevo paquete, se eliminaba el paquete con más antigüedad en el buffer, y se perdía sin ser procesado.

El tercer paso, y antes de implementar el protocolo robusto, se añadió un tiempo de espera antes de enviar un paquete, para no saturar el buffer de mensajes pendientes de ser procesados.

El cuarto paso, fue implementar el protocolo robusto, y que el waspmote receptor avisara con un mensaje de confirmación, que estaba preparado para seguir recibiendo mensajes.

4.6. Programación remota y mula de datos en waspmotes

Utilizando como base el protocolo robusto, se han añadido nuevas funcionalidades a la API de Libelium. Así, cuando un waspmote que haya sido programado como mula de datos entre en zona de cobertura de otro waspmote, automáticamente actuará de mula de datos, le entregará los datos que sean para ese waspmote, y solicitará al waspmote remoto que le envíe los datos que ha recolectado. Además, si fuese necesario, podrá cargar programas remotamente al otro waspmote y, una vez finalice la carga, podrá solicitar que ejecute el programa que acaba de cargar.

Hasta ahora, siempre era necesaria la utilización de un ordenador para realizar la programación remota, además de la interacción humana para realizar las selecciones en el shell. Con esta nueva funcionalidad, podremos programar al waspmote para que de manera desasistida realice las tareas.

5. Metodología

Para determinar la robustez del protocolo implementado, se han realizado pruebas de programación remota con el envío de dos programas de diferente tamaño. En nuestro caso, no es significativa la finalidad del programa, lo importante es el tamaño, que aumenta de manera directamente proporcional el número de paquetes necesarios para enviar ese programa.

Programa	Tamaño en bytes
A	1.890
B	65.564

Tabla 8: Tamaño de los programas utilizados en las pruebas

Para cada uno de los programas, se intentará realizar una programación remota, entre un ordenador con el shell desarrollado en Linux, y un waspmote a diferentes distancias, potencias y tamaños de paquete.

Distancias (metros)
100
500
1.000
2.000

Tabla 9: Distancias entre módulos XBee-PRO 868

5. Metodología

Potencia (mW)
1
25
100
200
300

Tabla 10: Potencia módulos XBee-PRO 868

El waspmote que recibirá el programa, está configurado con la versión modificada del API que implementa el nuevo protocolo robusto.

Todos los módulos XBee-PRO 868, tendrán la misma configuración de potencia, ambos módulos a 1mW, ambos a 25mW, etc., (ver tabla 10), y estarán equipados con una antena omnidireccional de 4.5dBi de ganancia. El puerto de comunicación se configurará a 38400 baudios, ocho bits de datos, un bit de parada, sin control de flujo y sin bit de paridad.

El escenario es el mismo para las pruebas, desde el mismo punto origen, vamos alejando el waspmote destino, según la tabla 9 de distancias, para que las condiciones de obstáculos sean las mismas.

Las antenas tendrán visión directa y cumplirán la regla del Elipsoide de Fresnel [28]. Se llama zona de Fresnel al volumen de espacio entre el emisor de una onda y el receptor, de modo que el desfase de las ondas en dicho volumen no supere los 180°.

5. Metodología

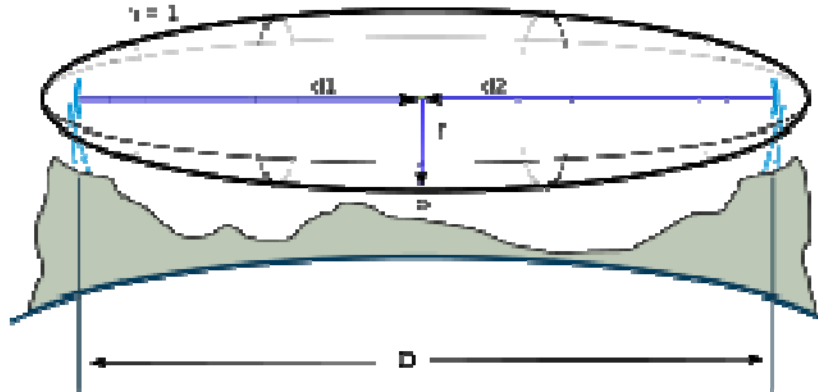


Figura 12: Elipsoide de Fresnel

El radio del elipsoide en su punto medio, viene determinado por la fórmula de la figura 13.

$$r_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}$$

Figura 13: Radio del Elipsoide de Fresnel

Donde r es el radio del elipsoide, d1 y d2 se consideran iguales para referenciar el punto medio, n será 1 para calcular la primera zona de Fresnel y L la longitud de onda de la señal. Los radios del elipsoide para cada una de las distancias de nuestro experimento, se representan en la tabla 11.

Distancia entre emisor y receptor (metros)	Radio del elipsoide (metros)
100	2.938
500	6.570
1.000	9.292
2.000	13.141

Tabla 11: Radio de la primera zona del Elipsoide de Fresnel para 868MHz

Lo ideal es que el Elipsoide de Fresnel esté libre de obstáculos, de no ser posible, se recomienda que los obstáculos no ocupen el 20% del volumen total del elipsoide.

5. Metodología

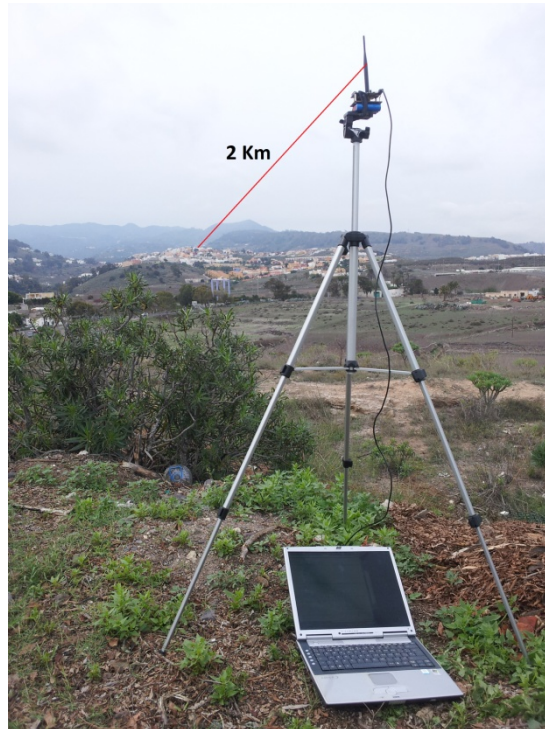


Figura 14: Wasp mote destino a 2 Km sin obstáculos

Además, para evitar posibles anomalías, debido a factores externos que podrían distorsionar los resultados, como pueden ser las condiciones meteorológicas y los ruidos que podrían afectar a nuestra señal de radiofrecuencia, los siguientes pasos han sido repetidos cuatro veces a razón de dos por día:

1. Configurar los módulos XBee-PRO 868 a la potencia más baja. Ver Tabla 10.
2. Separar los módulos XBee-PRO 868 a la distancia más corta. Ver Tabla 9.
3. Realizar la programación remota con el programa A. Ver Tabla 8.
 - Con todos los tamaños de carga útil del paquete de la tabla 12.
4. Realizar la programación remota con el programa B. Ver Tabla 9.
 - Con todos los tamaños de carga útil del paquete de la tabla 12.
5. Repetir paso 1, con la siguiente potencia hasta llegar a 300mW.
6. Evaluar los resultados.

5. Metodología

Carga útil del paquete (bytes)	Número de paquetes para el programa A	Número de paquetes para el programa B
72	29	979
80	26	875
90	23	772
94	22	737
100	20	691

Tabla 12: Tamaño de la carga útil (payload) del paquete y número de paquetes según programa

Estos factores, que no controlamos, generarán errores aleatorios en nuestras pruebas y ayudarán a determinar cómo se comporta y se recupera el protocolo ante múltiples imprevistos.



Figura 15: Wasp mote emisor durante el experimento

5. Metodología

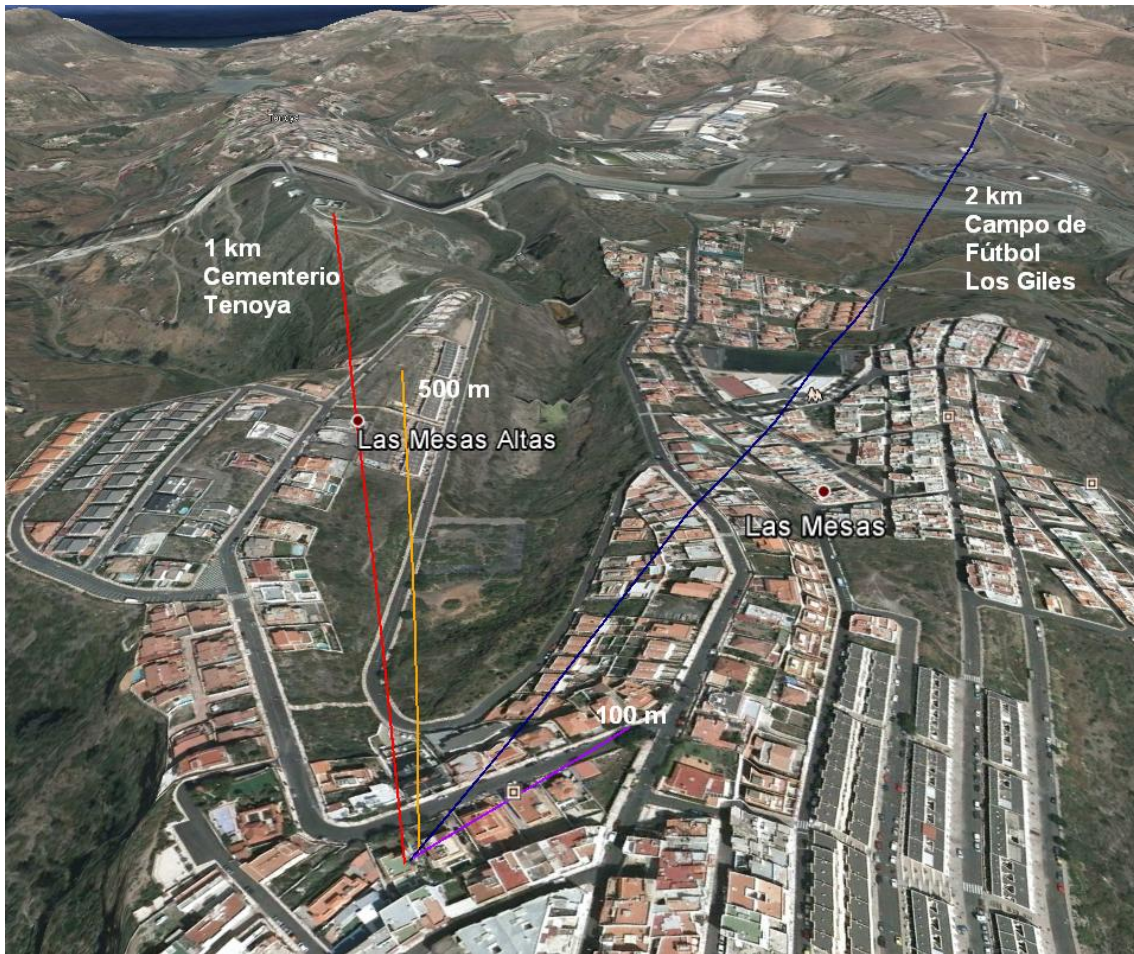


Figura 16: Mapa de Google indicando la situación del emisor y del receptor durante el experimento

6. Resultados

El número de paquetes necesarios para poder enviar completamente cada uno de los programas, será 34 veces superior para el programa B, respecto al programa A, según la tabla 12. Los resultados deberían normalizarse en el programa B, al ser un número tan elevado de paquetes.

6.1. Shell OTAP

Los dos programas han sido transmitidos correctamente para todas las distancias del experimento, tabla 9, y para todas las potencias admitidas por el módulo XBee-PRO 868, tabla 10.

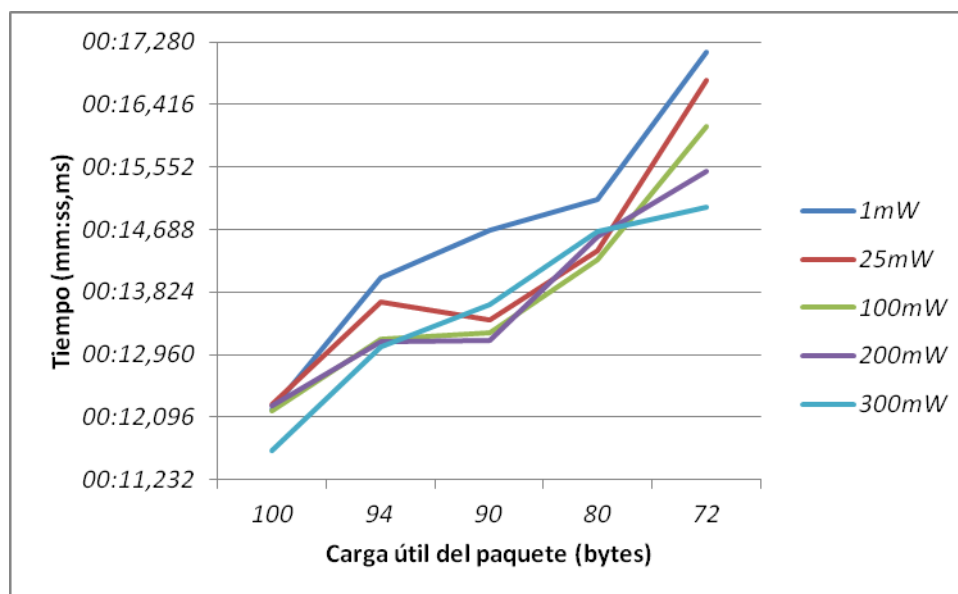


Figura 17: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa A con el shell OTAP

Como se observa en la figura 17 y 20, el tiempo medio de recepción del programa completo aumenta hasta un 38% si, en vez de utilizar la carga útil máxima de 100 bytes, utilizamos 72 bytes. En programas pequeños, como el programa A, estamos hablando de segundos, pero para el programa B, pasamos de 6 minutos a 8 minutos al aumentar el número de paquetes un 40%.

6. Resultados

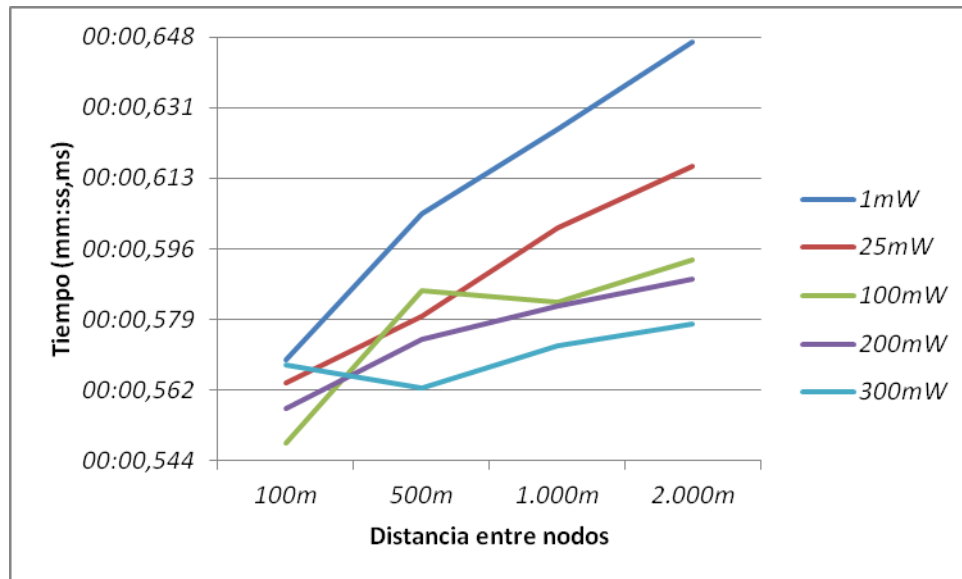


Figura 18: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A con el Shell OTAP

En las figuras 18 y 21 se representa el tiempo medio de recepción, procesamiento y confirmación por paquete que se envía por cada programa. Esto comprende, desde que se envía un paquete desde el emisor, hasta que lo recibe el receptor, lo procesa, lo almacena en la tarjeta SD y envía de vuelta una confirmación al emisor. En ambas figuras se aprecia que, en 100 metros, los resultados a la máxima potencia, 300mW, son peores que con otras potencias de emisión menores. Este es un problema conocido, denominado intermodulación [12] y que se produce al emitir a una excesiva potencia cuando las distancias de enlace entre el emisor y el receptor son relativamente cortas. La diferencia no es mayor a un 5% entre la potencia máxima y la mínima, para la distancia máxima entre nodos, de 2.000 metros.

6. Resultados

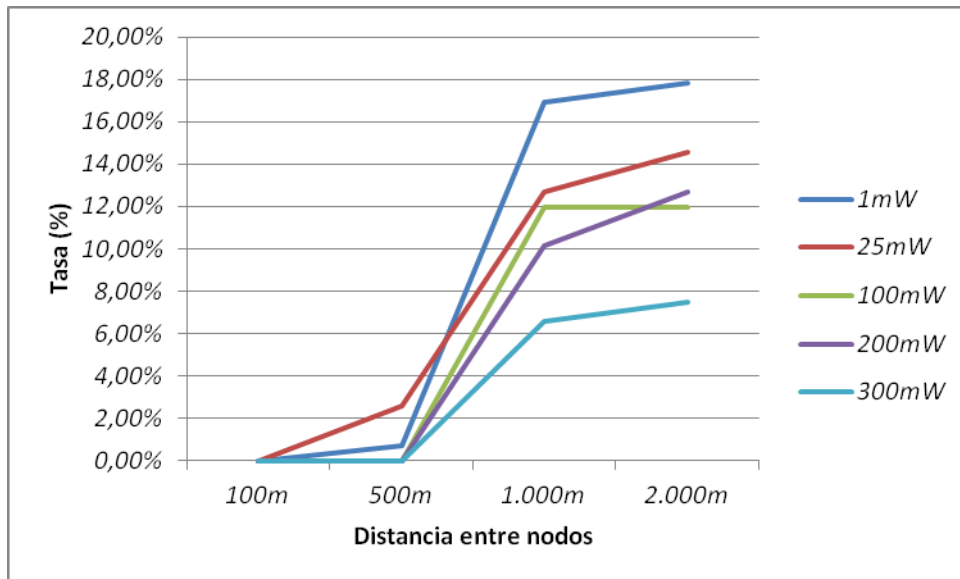


Figura 19: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A con el Shell OTAP

La tasa media de paquetes enviados aumenta a medida que aumenta la distancia entre los nodos, pero este aumento se atenúa aumentando la potencia del módulo XBee. Como se observa en la figura 21, utilizar la máxima potencia, disminuye en 25% la tasa de reenvíos.

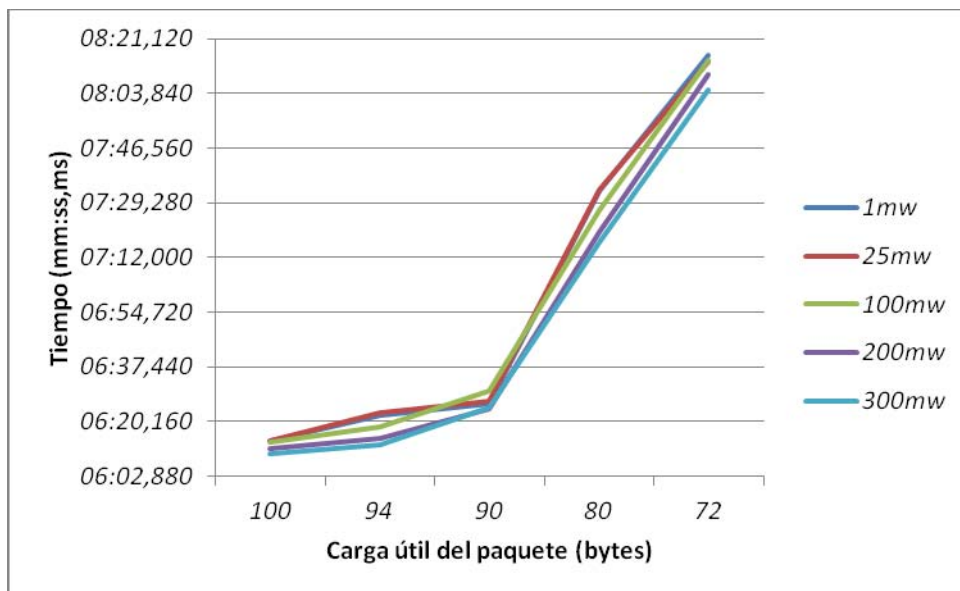


Figura 20: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa B con el shell OTAP

6. Resultados

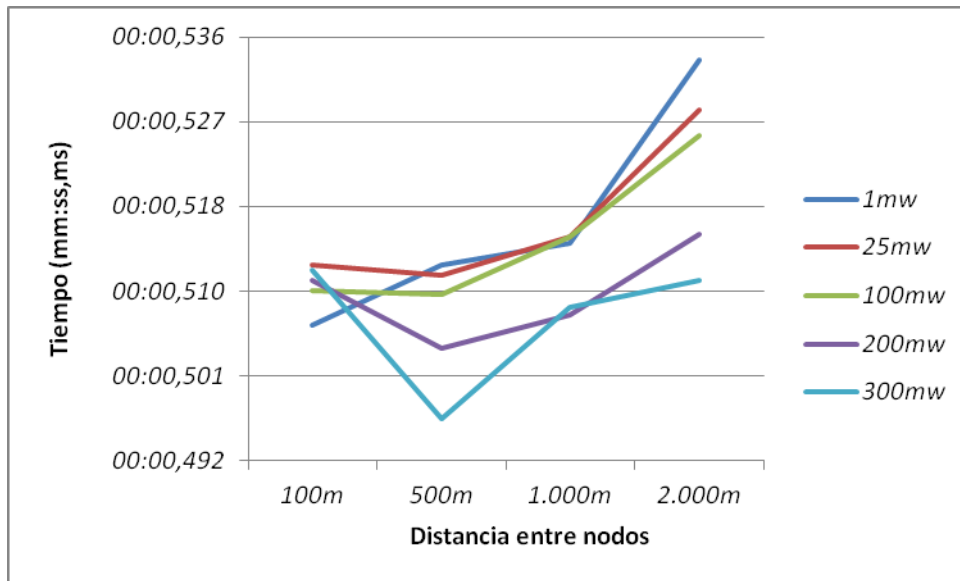


Figura 21: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B con el shell OTAP

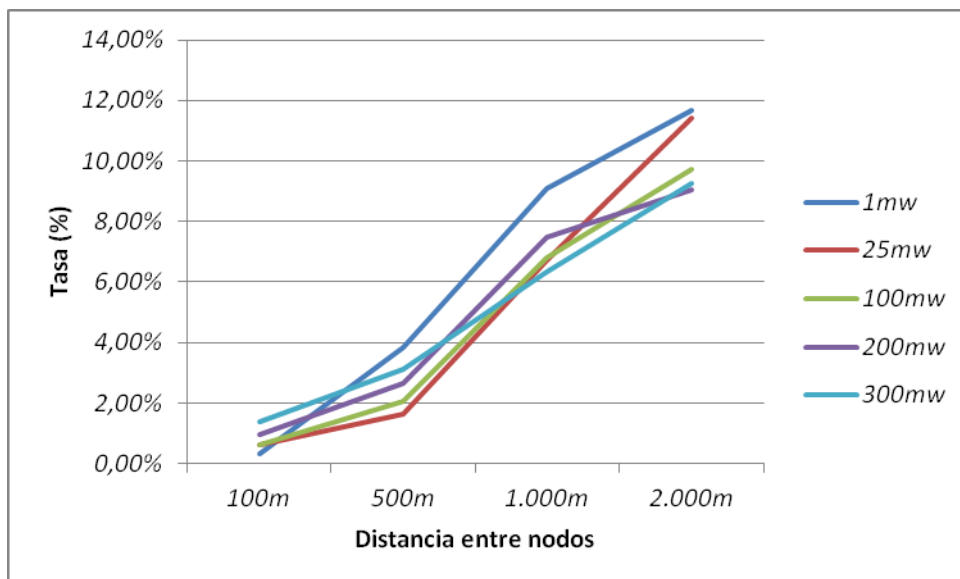


Figura 22: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B con el Shell OTAP

6. Resultados

Hemos comprobado cómo se comporta el ciclo de carga máxima del módulo XBee, en la figura 23 se muestra cómo se alcanza en una hora el 100% del ciclo de carga máxima, para ello hemos simulado el envío de un programa de tamaño infinito con la potencia de los módulos XBee-PRO 868 configuradas a 1mW. Una vez alcanzado el límite el módulo corta las comunicaciones, a partir de ese momento no se realizarán más envíos, sólo lecturas del porcentaje de ciclo de carga cada dos minutos y medio para comprobar cómo desciende.

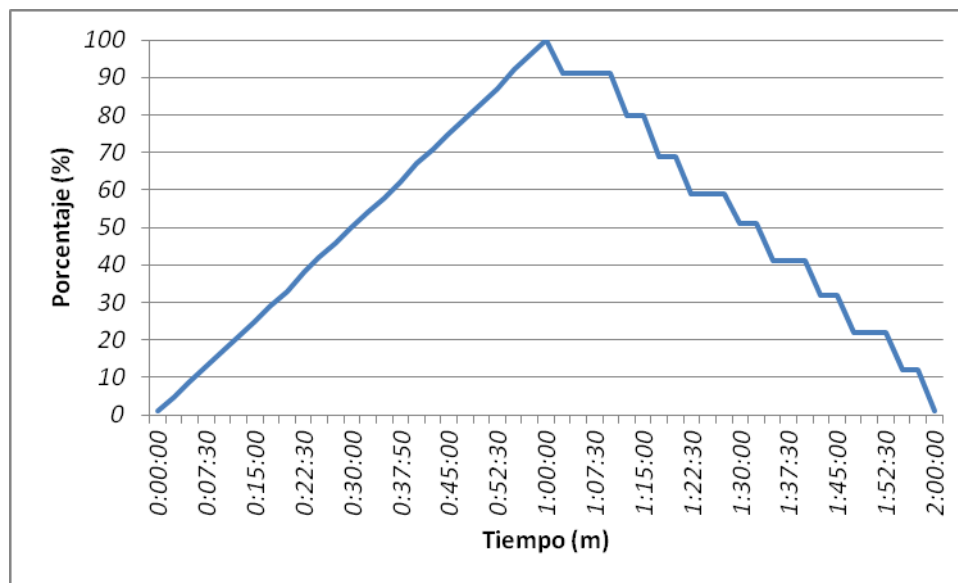


Figura 23: Ciclo de carga máxima: carga y descarga según tiempo

A pesar de que la carga se produce linealmente, la descarga es escalonada, aunque mantiene cierta simetría respecto a la carga, como se muestra en la figura 24, donde hemos puesto en espejo la línea recta de la carga para compararla con la descarga.

6. Resultados

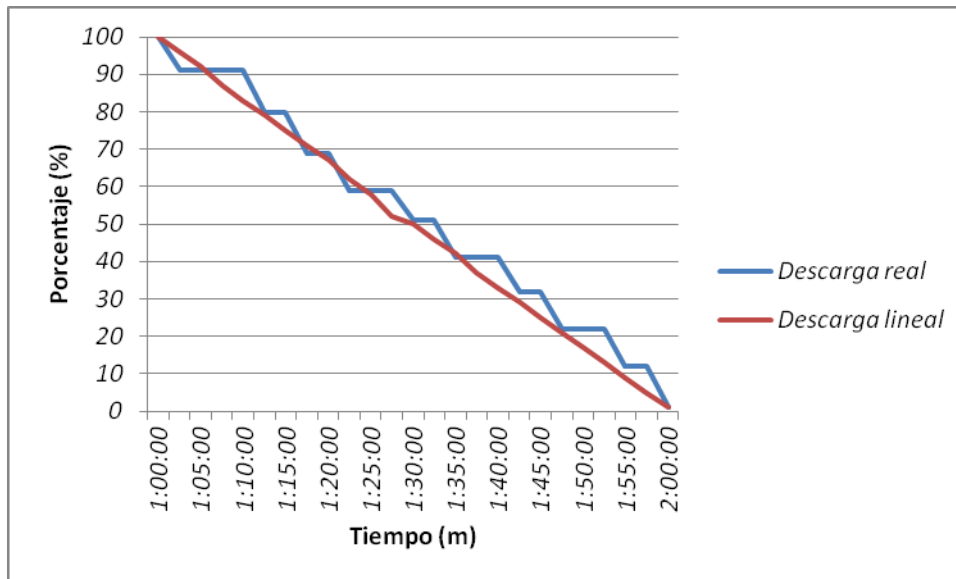


Figura 24: Descarga del ciclo de carga máxima según tiempo

El ciclo de carga máximo está limitado por dos factores, en primer lugar el estándar ETSI EN 300 220-1 [14], y en segundo lugar proteger el hardware para no alcanzar la temperatura máxima de 60°C. La temperatura de un módulo XBee-PRO 868 aumenta a medida que envía mensajes, alcanza rápidamente los 38°C y se termina estabilizando en 48°C hasta que alcanza el límite de ciclo de carga máximo.

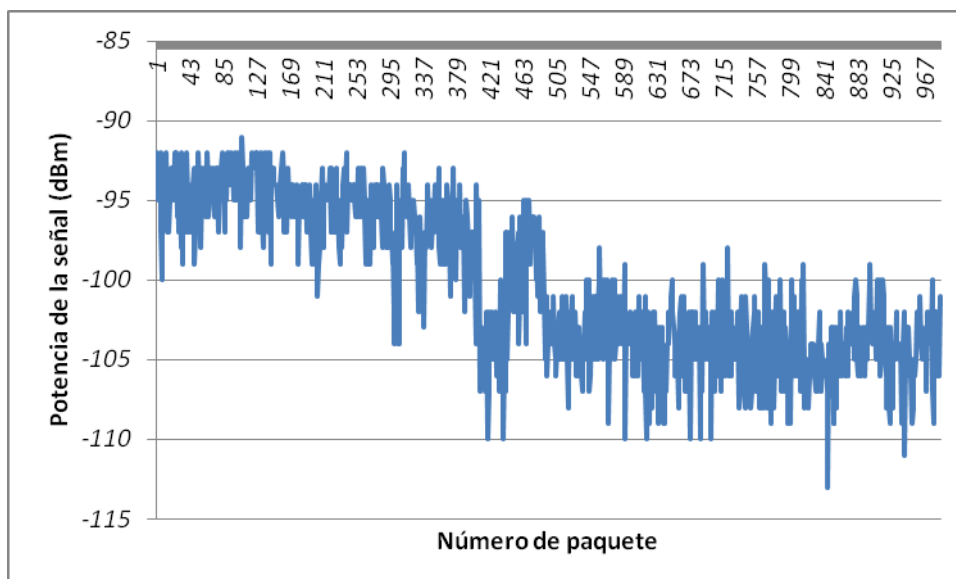


Figura 25: Potencia de la señal recibida por el módulo Xbee-PRO 868 a 25mW y a 100m de distancia

6. Resultados

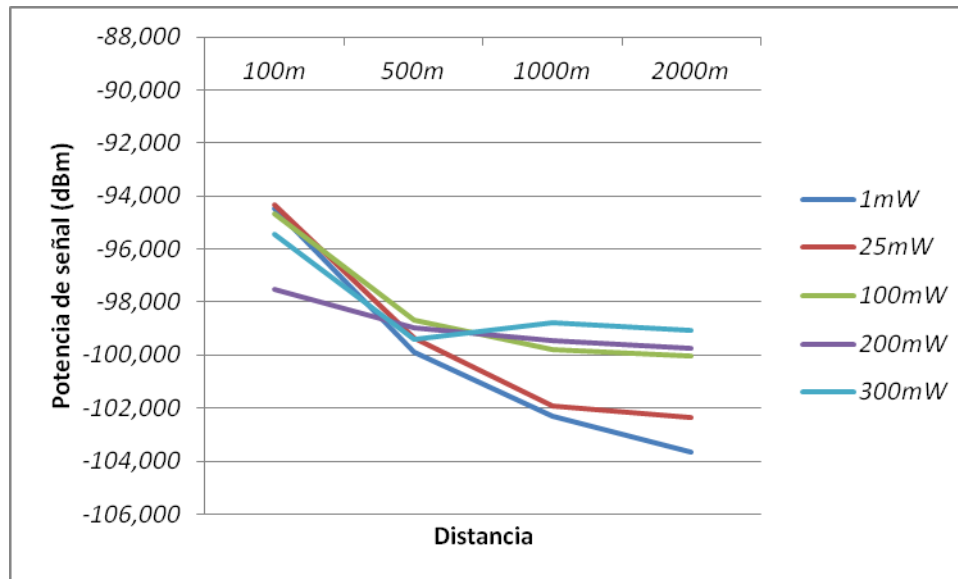


Figura 26: Potencia de la señal recibida según distancia y potencia del módulo XBee-PRO 868

Como se podía deducir de los resultados de tiempos medios de recepción completa, la potencia de la señal recibida cuando los módulos XBee-PRO 868 emiten a 1 y 25mW, se atenúa más a mayor distancia que con las potencias 100, 200 y 300mW.

6.2. OTAP entre waspmotes

La programación remota entre waspmotes se ha probado sólo con el programa A. En la figura 28 se observa el tiempo medio de recepción, procesamiento y confirmación, la potencia máxima de 300mW mejora a medida que nos alejamos del nodo, y si estamos muy cerca, las mayores potencias se ven afectadas por la intermodulación.

6. Resultados

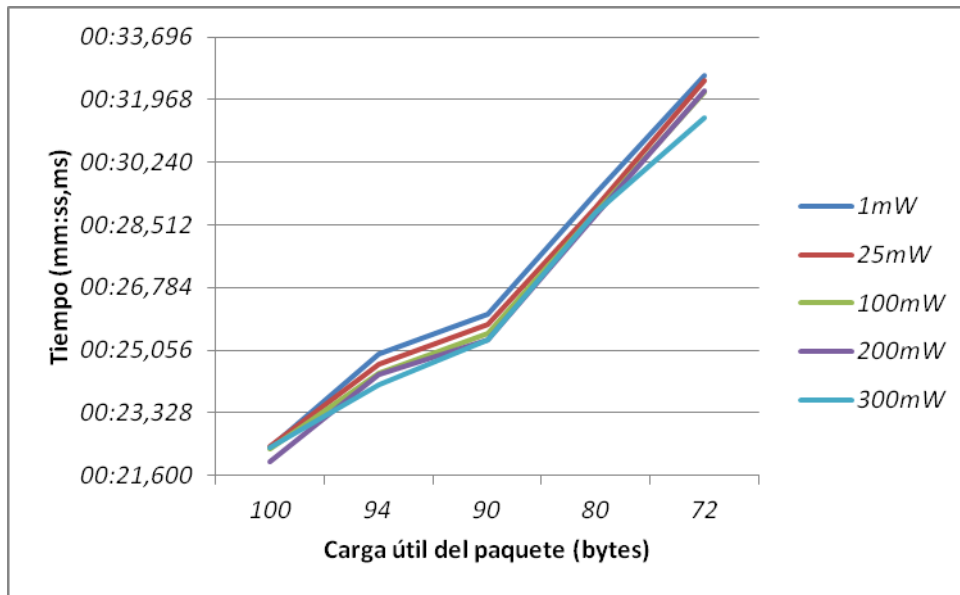


Figura 27: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa A con el shell OTAP

El tiempo medio ha aumentado el doble respecto al tiempo medio del shell OTAP. El hecho de tener otro waspmotes ralentiza el proceso:

- El waspmote emisor debe procesar el fichero con el programa a enviar, para el programa A significa unos 23sg y 457ms antes de poder enviar el primer paquete.
- Enviar el paquete.
- Esperar a recibir la confirmación.
- Al recibir la confirmación se envía el siguiente paquete, tiene que preparar el nuevo trozo de programa que debe empaquetar para enviar. Esta es la otra parte que ralentiza que el tiempo final sea mayor al shell OTAP.

6. Resultados

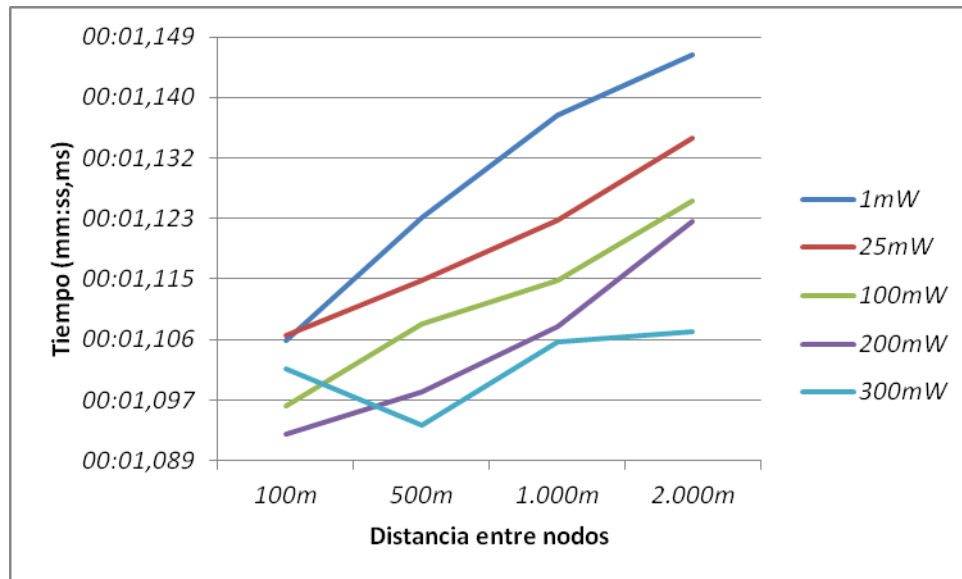


Figura 28: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A entre waspmotes

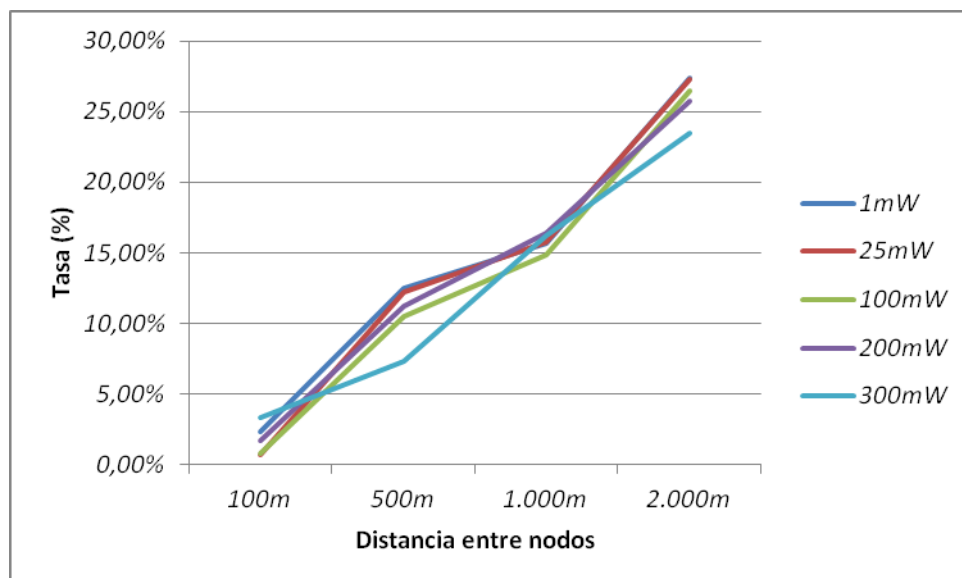


Figura 29: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A entre waspmotes

La tasa de fallos ha aumentado respecto al shell OTAP, pero a pesar de ello los programas se han transmitido correctamente. Existe un tiempo de reenvío que si no se recibe confirmación el paquete se vuelve a enviar, estos reenvíos contabilizan como errores, debemos asegurarnos de configurar bien el tiempo

6. Resultados

de reenvío cuando realicemos operaciones entre waspmotes para no provocar errores. Este tiempo de reenvío deberá ser mayor al aumentar el tiempo medio por paquete.

6.3. Shell OTAP primera versión

La primera prueba de concepto que implementamos del shell OTAP para que funcionara igual que el shell OTAP de Libelium, se basaba en realizar retardos para permitir al waspmote que procesara el paquete que enviábamos.

Existían tres puntos de retardo:

- Después del paquete que indica comienzo de OTAP. Retardo de 1.5sg.
- Después de cada paquete enviado con una parte del código del programa. Retardo de 0.5sg.
- Después del paquete de finalización de OTAP. Retardo de 2sg.

La decisión de cómo fijar cada uno de los retardos fue la experiencia realizando pruebas, era el tiempo necesario para que la programación remota funcionara sin problemas.

Sumando los retardos para 29 paquetes que consta el programa A para tamaño 72, tenemos un total de 18sg.

A 100 metros para tamaño 72 el shell OTAP del protocolo robusto tarda 16.015sg a diferencia de los 18.320sg que tarda el shell OTAP primera versión. Tarda menos la versión final del shell OTAP, aunque es cierto que envía el doble de paquetes, ya que se ha sustituido los retardos por paquetes de confirmación.

7. Conclusiones

El objetivo principal de este Trabajo Fin de Máster era el desarrollo de un protocolo de comunicación robusto punto a punto que gestionara el intercambio de un volumen elevado de paquetes entre dos waspmotes aislados pertenecientes a una red de sensores.

Las limitaciones hardware, tanto del waspmote en procesamiento y capacidad como del módulo XBee-PRO 868 en ciclo de carga máxima han condicionado el diseño del protocolo.

El abanico de servicios creados en torno a este protocolo robusto dota a las redes sensoriales de nuevas capacidades, ampliando las aplicaciones que pueden desarrollarse sobre estos dispositivos.

Los resultados de las pruebas experimentales realizadas demuestran que se ha conseguido una comunicación robusta basada en módulos de radio frecuencia XBee. Este protocolo se ha ensayado entre un portátil y una mota, y entre dos motas para soportar operaciones OTAP y de transporte de datos (data muling). En todos los casos los resultados han sido satisfactorios, reponiéndose de errores de timeout y de fallos de recepción de paquetes que de otra forma hubiera supuesto la pérdida de conexión y la necesidad de reiniciar la transmisión desde el primer paquete.

7. Conclusiones

7.1. Líneas futuras

A continuación se enumeran líneas futuras de investigación a partir de este trabajo:

- OTAP Multipunto: una línea interesante de desarrollo es la extensión del protocolo para permitir una comunicación multipunto y difusión. Se debería articular de tal forma que gestione los ciclos de carga máxima de cada una de los módulos XBee-PRO 868 de la red. Esta posibilidad permitiría desarrollar nuevos e interesantes servicios OTAP.
- Implementar un protocolo que utilice una ventana de transmisión para el control de errores, reduciendo el número de confirmaciones y el tiempo de comunicación total. El protocolo gestionará el buffer de mensajes pendientes que ya dispone el API de Libelium para simular una ventana deslizante, confirmar el paquete n implicaría confirmar todos los anteriores sin necesidad de enviar una confirmación por paquete.
- Desarrollar un protocolo energéticamente más eficiente, que regule la potencia de transmisión de las radios según la distancia y calidad de la señal a la que se encuentren para minimizar el consumo energético.

Referencias

- [1] Artículo de redes sensoriales de Libelium.
http://www.libelium.com/es/smart_cities_wsn_overview/
- [2] Empresa Libelium. <http://www.libelium.com>
- [3] Proyecto Arduino, <http://www.arduino.cc/es/>
- [4] Módulo XBee-PRO 868. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xbee/>
- [5] Empresa Digi. <http://www.digi.com/es/>
- [6] Especificaciones del módulo XBee-PRO 868. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-pro-868>
- [7] David Gascón, Alberto Bielsa, Marco Zennaro y Antoine Bagula. "Poster Abstract: Experimental Evaluation of Radio Transceivers for Sensor Networks in Harsh Environments". European Wireless Sensor Network Conference. Portugal. 2010.
<http://www.libelium.com/libelium-downloads/EWSN2010PosterFinal.pdf>
- [8] Tienda online de productos para waspmote de Libelium y Arduino, <http://www.cooking-hacks.com>
- [9] Descarga de X-CTU. <http://www.digi.com/support/productdetail?pid=3352>
- [10] Manual de operación y configuración de Módulos XBee-PRO 868 de Digi. "XBee-PRO 868 RF Modules". 2011.
- [11] Manual sobre programación remota en waspmotes de Libelium. "Over the Air Programming with 802.15.4 and ZigBee". v0.9. Libelium Distributed Communications. España. 07/2012. http://www.libelium.com/v11-files/documentation/waspmote/over_the_air_programming.pdf
- [12] Definición del concepto de intermodulación.
<http://es.wikipedia.org/wiki/Intermodulaci%C3%B3n>
- [13] Definición de Mula de Datos. http://en.wikipedia.org/wiki/Data_mule

- [14] Estándar ETSI EN 300 220-1 <http://www.rfm.com/corp/new868dat/cept.pdf>
- [15] Definición de Internet de las cosas. http://es.wikipedia.org/wiki/Internet_de_las_Cosas
- [16] Matthew Dunbabin, Peter Corke, Iuliu Vasilescu y Daniela Rus. "Data Muling over Underwater Wireless Sensor Networks using an Autonomous Underwater Vehicle". Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Florida. 2006.
- [17] Kamal Beydoun, Violeta Felea. "Energy-Efficient WSN Infrastructure". International Symposium on Collaborative Technologies and Systems. 2008.
- [18] Stefano Basagni, Michele Nati, Chiara Petrioli y Roberto Petrocchia. "ROME: Routing Over Mobile Elements in WSNs". IEEE GLOBECOM 2009.
- [19] Stefan Unterschütz y Volker Turau. "Fail-Safe Over-The-Air Programming and Error Recovery in Wireless Networks" in Proceeding of Tenth Workshop on Intelligent Solutions in Embedded Systems (WISES). 2012.
- [20] Rashmi Parthasarathy, Nina Peterson, WenZhan Song, Ali Hurson y Behrooz A. Shirazi. "Over the Air Programming on Imote2-based Sensor Networks". IEEE Computer Society. 2010.
- [21] Marco Zennaro, Antoine Bagula y David Gascón. "Long Distance Wireless Sensor Networks: simulation vs reality". Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions. California. 2010.
- [22] Foro de Libelium sobre OTAP para la version 1.1 de waspmote. <http://www.libelium.com/forum/viewforum.php?f=19>
- [23] David Jea, Arun Somasundara y Mani Srivastava. "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks". Springer. 2005.
- [24] Definición del Problema del Viajero. http://es.wikipedia.org/wiki/Problema_del_viajante
- [25] Yu-Chee Tseng, Fang-Jing Wu, Wan-Ting Lai. "Opportunistic data collection for disconnected wireless sensor networks by mobile mules". "Ad Hoc Network". Páginas 1150-1164. Elsevier. 2013
- [26] F. Gebali. "Analysis of Computer and Communication Networks". Capítulo 9: "Modeling Error Control Protocols". Springer Science+Business Media. 2008.

[27] Librería C/C++ LibXBee. V3.09. <https://code.google.com/p/libxbee/>

[28] Definición de Zona de Fresnel. http://es.wikipedia.org/wiki/Zona_de_Fresnel

Índice de figuras

Figura 1: Waspote: mota de Libelium	6
Figura 2: módulo XBee-PRO 868 de Digi	7
Figura 3: Adaptadores USB para XBee [8]	8
Figura 4: Programa X-CTU de Digi para programar los módulos XBee	8
Figura 5: Modo de transmisión unidifusión (unicast)	10
Figura 6: Modo de transmisión multidifusión (multicast).....	10
Figura 7: Modo de transmisión difusión (broadcast).....	10
Figura 8: Protocolo parada y espera	26
Figura 9: Protocolo vuelta atrás N.....	27
Figura 10: Protocolo repetición selectiva.....	27
Figura 11: Shell OTAP Libelium.....	30
Figura 12: Elipsoide de Fresnel.....	34
Figura 13: Radio del Elipsoide de Fresnel.....	34
Figura 14: Waspote destino a 2 Km sin obstáculos	35
Figura 15: Waspote emisor durante el experimento.....	36
Figura 16: Mapa de Google indicando la situación del emisor y del receptor durante el experimento	37
Figura 17: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa A con el shell OTAP	38
Figura 18: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A con el Shell OTAP	39
Figura 19: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A con el Shell OTAP	40
Figura 20: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa B con el shell OTAP	40

Figura 21: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B con el shell OTAP.....	41
Figura 22: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B con el Shell OTAP	41
Figura 23: Ciclo de carga máxima: carga y descarga según tiempo.....	42
Figura 24: Descarga del ciclo de carga máxima según tiempo.....	43
Figura 25: Potencia de la señal recibida por el módulo Xbee-PRO 868 a 25mW y a 100m de distancia	43
Figura 26: Potencia de la señal recibida según distancia y potencia del módulo XBee-PRO 86844	
Figura 27: Tiempo medio de recepción completa según el tamaño de la carga útil (payload) del paquete y potencia del módulo XBee-PRO 868 para el programa A con el shell OTAP	45
Figura 28: Tiempo medio de recepción, procesamiento y confirmación por paquete según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A entre waspmotes	46
Figura 29: Tasa media de paquetes reenviados según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A entre waspmotes	46

Índice de tablas

Tabla 1: Comparativa módulos XBee de Digi	22
Tabla 2: Estructura general de una trama XBee de Digi.....	23
Tabla 3: Estructura de trama API para el envío de datos a otro módulo.....	23
Tabla 4: Estructura del campo datos de un paquete OTAP de Libelium.....	24
Tabla 5: Estructura del campo datos de una trama OTAP para el envío de un programa.....	25
Tabla 6: Códigos confirmación de trama OTAP.....	28
Tabla 7: Códigos de trama y de confirmación de trama para mula de datos	29
Tabla 8: Tamaño de los programas utilizados en las pruebas.....	32
Tabla 9: Distancias entre módulos XBee-PRO 868	32
Tabla 10: Potencia módulos XBee-PRO 868	33
Tabla 11: Radio de la primera zona del Elipsoide de Fresnel para 868MHz	34
Tabla 12: Tamaño de la carga útil (payload) del paquete y número de paquetes según programa	36

Apéndice 1

A.1. Shell OTAP - Programa A

Resultados programa A, según distancia entre nodos, potencia del módulo XBee-PRO 868 y tamaño de la carga útil del paquete

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	00:11,663	00:11,622	00:11,196	00:10,965	00:11,447
500 m	00:12,265	00:12,349	00:12,889	00:12,524	00:11,813
1.000 m	00:12,418	00:12,360	00:12,233	00:12,670	00:11,608
2.000 m	00:12,458	00:12,760	00:12,430	00:12,870	00:11,668

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	00:12,986	00:13,003	00:12,306	00:12,912	00:13,209
500 m	00:13,985	00:13,256	00:12,987	00:13,132	00:12,875
1.000 m	00:14,539	00:14,134	00:13,520	00:13,250	00:13,227
2.000 m	00:14,570	00:14,348	00:13,879	00:13,264	00:12,986

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	00:13,122	00:12,502	00:12,609	00:12,912	00:13,888
500 m	00:14,672	00:13,250	00:13,899	00:13,100	00:13,340
1.000 m	00:15,125	00:13,777	00:12,973	00:13,321	00:13,346
2.000 m	00:15,799	00:14,250	00:13,568	00:13,254	00:14,002

Programa A					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	00:14,264	00:13,928	00:13,920	00:14,596	00:14,189
500 m	00:14,987	00:13,925	00:14,008	00:14,456	00:14,280
1.000 m	00:15,342	00:14,712	00:14,501	00:14,505	00:15,020
2.000 m	00:15,815	00:14,981	00:14,622	00:14,788	00:15,122

Programa A					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	00:16,015	00:16,397	00:15,666	00:15,285	00:14,960
500 m	00:16,232	00:16,445	00:15,986	00:15,111	00:14,676
1.000 m	00:17,344	00:16,865	00:16,477	00:15,587	00:15,133
2.000 m	00:18,943	00:17,300	00:16,301	00:16,003	00:15,227

Anexo 1

Media de tiempo de recepción completo del programa A, según tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868

Programa A	100	94	90	80	72
1mW	00:12,201	00:14,020	00:14,680	00:15,102	00:17,134
25mW	00:12,273	00:13,685	00:13,445	00:14,387	00:16,752
100mW	00:12,187	00:13,173	00:13,262	00:14,263	00:16,108
200mW	00:12,257	00:13,140	00:13,147	00:14,586	00:15,497
300mW	00:11,634	00:13,074	00:13,644	00:14,653	00:14,999

Tiempo medio de recepción y procesamiento por paquete según distancia entre nodos, tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,583	00:00,581	00:00,560	00:00,548	00:00,572
500 m	00:00,613	00:00,617	00:00,644	00:00,626	00:00,591
1.000 m	00:00,621	00:00,618	00:00,612	00:00,634	00:00,580
2.000 m	00:00,623	00:00,638	00:00,622	00:00,644	00:00,583

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,590	00:00,591	00:00,559	00:00,587	00:00,600
500 m	00:00,636	00:00,603	00:00,590	00:00,597	00:00,585
1.000 m	00:00,661	00:00,642	00:00,615	00:00,602	00:00,601
2.000 m	00:00,662	00:00,652	00:00,631	00:00,603	00:00,590

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,571	00:00,544	00:00,548	00:00,561	00:00,604
500 m	00:00,638	00:00,576	00:00,604	00:00,570	00:00,580
1.000 m	00:00,658	00:00,599	00:00,564	00:00,579	00:00,580
2.000 m	00:00,687	00:00,620	00:00,590	00:00,576	00:00,609

Programa A					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,549	00:00,536	00:00,535	00:00,561	00:00,546
500 m	00:00,576	00:00,536	00:00,539	00:00,556	00:00,549
1.000 m	00:00,590	00:00,566	00:00,558	00:00,558	00:00,578
2.000 m	00:00,608	00:00,576	00:00,562	00:00,569	00:00,582

Programa A					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,552	00:00,565	00:00,540	00:00,527	00:00,516
500 m	00:00,560	00:00,567	00:00,551	00:00,521	00:00,506
1.000 m	00:00,598	00:00,582	00:00,568	00:00,537	00:00,522
2.000 m	00:00,653	00:00,597	00:00,562	00:00,552	00:00,525

Media del tiempo medio de recepción y procesamiento por paquete, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,569	00:00,563	00:00,549	00:00,557	00:00,568
500 m	00:00,605	00:00,580	00:00,586	00:00,574	00:00,562
1.000 m	00:00,626	00:00,601	00:00,583	00:00,582	00:00,572
2.000 m	00:00,647	00:00,616	00:00,593	00:00,589	00:00,578

Anexo 1

Número de reenvíos y tasa de reenvíos, según tamaño de la carga útil del paquete, distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	0	1	0	0	0
1.000 m	4	3	3	2	1
2.000 m	4	4	3	3	2

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	0,00%	5,00%	0,00%	0,00%	0,00%
1.000 m	20,00%	15,00%	15,00%	10,00%	5,00%
2.000 m	20,00%	20,00%	15,00%	15,00%	10,00%

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	0	0	0	0	0
1.000 m	4	3	3	2	1
2.000 m	4	4	3	3	1

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	0,00%	0,00%	0,00%	0,00%	0,00%
1.000 m	18,18%	13,64%	13,64%	9,09%	4,55%
2.000 m	18,18%	18,18%	13,64%	13,64%	4,55%

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	0	1	0	0	0
1.000 m	4	3	3	3	2
2.000 m	5	3	3	3	2

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	0,00%	4,35%	0,00%	0,00%	0,00%
1.000 m	17,39%	13,04%	13,04%	13,04%	8,70%
2.000 m	21,74%	13,04%	13,04%	13,04%	8,70%

Programa A					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	0	0	0	0	0
1.000 m	4	3	2	3	2
2.000 m	4	2	2	3	1

Programa A					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	0,00%	0,00%	0,00%	0,00%	0,00%
1.000 m	15,38%	11,54%	7,69%	11,54%	7,69%
2.000 m	15,38%	7,69%	7,69%	11,54%	3,85%

Programa A					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	1	1	0	0	0
1.000 m	4	3	3	2	2
2.000 m	4	4	3	3	3

Programa A					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	3,45%	3,45%	0,00%	0,00%	0,00%
1.000 m	13,79%	10,34%	10,34%	6,90%	6,90%
2.000 m	13,79%	13,79%	10,34%	10,34%	10,34%

Anexo 1

Tasa media de reenvíos, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	0,69%	2,56%	0,00%	0,00%	0,00%
1.000 m	16,95%	12,71%	11,94%	10,11%	6,57%
2.000 m	17,82%	14,54%	11,94%	12,71%	7,49%

A.1. Shell OTAP - Programa B

Resultados programa B, según distancia entre nodos, potencia del módulo XBee-PRO 868 y tamaño de la carga útil del paquete

Programa B					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	05:50,927	06:03,237	06:03,395	06:04,967	06:03,586
500 m	06:18,657	06:05,444	06:07,634	06:08,778	06:04,673
1.000 m	06:15,943	06:19,485	06:18,321	06:12,113	06:15,321
2.000 m	06:30,854	06:27,432	06:25,368	06:20,533	06:16,720

Programa B					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	06:18,438	06:20,132	06:13,390	06:17,592	06:15,732
500 m	06:20,532	06:22,311	06:14,310	06:09,412	06:07,132
1.000 m	06:19,852	06:21,803	06:18,933	06:12,534	06:12,533
2.000 m	06:29,701	06:27,523	06:27,316	06:20,123	06:16,654

Programa B					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	06:30,375	06:35,095	06:45,883	06:43,600	06:45,071
500 m	06:20,599	06:22,931	06:22,412	06:18,459	06:17,132
1.000 m	06:20,870	06:20,144	06:23,013	06:15,123	06:17,940
2.000 m	06:30,512	06:28,123	06:27,850	06:19,311	06:18,333

Programa B					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	07:25,523	07:28,862	07:19,012	07:14,961	07:25,694
500 m	07:25,812	07:30,337	07:24,224	07:15,135	06:59,112
1.000 m	07:30,330	07:29,981	07:26,112	07:22,113	07:20,156
2.000 m	07:50,117	07:45,113	07:39,007	07:27,773	07:22,412

Programa B					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	08:04,792	08:05,437	07:59,008	08:03,462	07:58,983
500 m	08:05,119	08:08,005	08:13,664	08:06,788	08:00,001
1.000 m	08:15,133	08:12,030	08:17,980	08:11,001	08:09,136
2.000 m	08:37,840	08:29,506	08:27,166	08:17,662	08:11,776

Anexo 1

Media de tiempo de recepción completo del programa B, según tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868

Programa B	100	94	90	80	72
1mW	06:14,095	06:22,131	06:25,589	07:32,946	08:15,721
25mW	06:13,899	06:22,942	06:26,573	07:33,573	08:13,745
100mW	06:13,679	06:18,487	06:29,789	07:27,089	08:14,454
200mW	06:11,598	06:14,915	06:24,123	07:19,996	08:09,728
300mW	06:10,075	06:13,013	06:24,619	07:16,843	08:04,974

Tiempo medio de recepción y procesamiento por paquete según distancia entre nodos, tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868 para el programa B

Programa B					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,508	00:00,526	00:00,526	00:00,528	00:00,526
500 m	00:00,548	00:00,529	00:00,532	00:00,534	00:00,528
1.000 m	00:00,544	00:00,549	00:00,547	00:00,539	00:00,543
2.000 m	00:00,566	00:00,561	00:00,558	00:00,551	00:00,545

Programa B					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,513	00:00,516	00:00,507	00:00,512	00:00,510
500 m	00:00,516	00:00,519	00:00,508	00:00,501	00:00,498
1.000 m	00:00,515	00:00,518	00:00,514	00:00,505	00:00,505
2.000 m	00:00,529	00:00,526	00:00,526	00:00,516	00:00,511

Programa B					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,506	00:00,512	00:00,526	00:00,523	00:00,525
500 m	00:00,493	00:00,496	00:00,495	00:00,490	00:00,489
1.000 m	00:00,493	00:00,492	00:00,496	00:00,486	00:00,490
2.000 m	00:00,506	00:00,503	00:00,502	00:00,491	00:00,490

Programa B					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,509	00:00,513	00:00,502	00:00,497	00:00,509
500 m	00:00,509	00:00,515	00:00,508	00:00,497	00:00,479
1.000 m	00:00,515	00:00,514	00:00,510	00:00,505	00:00,503
2.000 m	00:00,537	00:00,532	00:00,525	00:00,512	00:00,506

Programa B					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,495	00:00,496	00:00,489	00:00,494	00:00,489
500 m	00:00,496	00:00,498	00:00,504	00:00,497	00:00,490
1.000 m	00:00,506	00:00,503	00:00,509	00:00,502	00:00,500
2.000 m	00:00,529	00:00,520	00:00,518	00:00,508	00:00,502

Media del tiempo medio de recepción y procesamiento por paquete, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B

Programa B					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	00:00,506	00:00,512	00:00,510	00:00,511	00:00,512
500 m	00:00,512	00:00,511	00:00,509	00:00,504	00:00,497
1.000 m	00:00,515	00:00,515	00:00,515	00:00,507	00:00,508
2.000 m	00:00,533	00:00,528	00:00,526	00:00,516	00:00,511

Anexo 1

Número de reenvíos y tasa de reenvíos, según tamaño de la carga útil del paquete, distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B

Programa B					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	1	5	3	10	16
500 m	32	11	16	26	30
1.000 m	70	65	55	63	61
2.000 m	97	106	72	80	77

Programa B					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	0,14%	0,72%	0,43%	1,45%	2,32%
500 m	4,63%	1,59%	2,32%	3,76%	4,34%
1.000 m	10,13%	9,41%	7,96%	9,12%	8,83%
2.000 m	14,04%	15,34%	10,42%	11,58%	11,14%

Programa B					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	2	1	1	2	8
500 m	25	15	10	20	21
1.000 m	66	61	60	66	51
2.000 m	78	82	80	58	63

Programa B					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	0,27%	0,14%	0,14%	0,27%	1,09%
500 m	3,39%	2,04%	1,36%	2,71%	2,85%
1.000 m	8,96%	8,28%	8,14%	8,96%	6,92%
2.000 m	10,58%	11,13%	10,85%	7,87%	8,55%

Programa B					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	1	3	7	11	9
500 m	33	9	20	15	22
1.000 m	69	30	44	55	37
2.000 m	91	88	76	71	80

Programa B					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	0,13%	0,39%	0,91%	1,42%	1,17%
500 m	4,27%	1,17%	2,59%	1,94%	2,85%
1.000 m	8,94%	3,89%	5,70%	7,12%	4,79%
2.000 m	11,79%	11,40%	9,84%	9,20%	10,36%

Programa B					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	3	7	1	5	11
500 m	33	17	16	22	23
1.000 m	88	46	51	55	43
2.000 m	99	86	75	69	68

Programa B					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	0,34%	0,80%	0,11%	0,57%	1,26%
500 m	3,77%	1,94%	1,83%	2,51%	2,63%
1.000 m	10,06%	5,26%	5,83%	6,29%	4,91%
2.000 m	11,31%	9,83%	8,57%	7,89%	7,77%

Programa B					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	8	10	15	10	10
500 m	31	15	21	22	28
1.000 m	72	66	62	59	62
2.000 m	105	93	88	85	83

Programa B					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	0,82%	1,02%	1,53%	1,02%	1,02%
500 m	3,17%	1,53%	2,15%	2,25%	2,86%
1.000 m	7,35%	6,74%	6,33%	6,03%	6,33%
2.000 m	10,73%	9,50%	8,99%	8,68%	8,48%

Anexo 1

Tasa media de reenvíos, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa B

Programa B					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	0,34%	0,61%	0,62%	0,95%	1,37%
500 m	3,85%	1,65%	2,05%	2,64%	3,11%
1.000 m	9,09%	6,71%	6,79%	7,50%	6,36%
2.000 m	11,69%	11,44%	9,74%	9,04%	9,26%

A.1. OTAP entre waspmotes - Programa A

Resultados programa A, según distancia entre nodos, potencia del módulo XBee-PRO 868 y tamaño de la carga útil del paquete

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	00:22,143	00:22,107	00:21,878	00:21,932	00:21,672
500 m	00:22,663	00:22,334	00:22,543	00:21,855	00:22,433
1.000 m	00:22,599	00:22,454	00:22,345	00:21,664	00:21,586
2.000 m	00:22,988	00:22,766	00:22,545	00:22,453	00:21,983

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	00:24,443	00:24,401	00:24,103	00:24,053	00:23,876
500 m	00:24,763	00:24,497	00:24,175	00:24,158	00:23,903
1.000 m	00:24,801	00:24,759	00:24,552	00:24,499	00:24,122
2.000 m	00:25,199	00:25,002	00:24,869	00:24,850	00:24,541

Programa A

90					
	1mw	25mw	100mw	200mw	300mw
100m	00:25,412	00:25,385	00:25,228	00:25,003	00:25,412
500m	00:25,660	00:25,690	00:25,332	00:25,211	00:24,995
1.000m	00:25,920	00:25,884	00:25,690	00:25,556	00:25,442
2.000m	00:26,337	00:26,078	00:25,842	00:25,582	00:25,532

Programa A

80					
	1mW	25mW	100mW	200mW	300mW
100 m	00:28,434	00:28,669	00:28,404	00:28,321	00:28,354
500 m	00:28,992	00:28,772	00:28,658	00:28,542	00:28,432
1.000 m	00:29,395	00:28,994	00:28,944	00:28,909	00:28,895
2.000 m	00:29,603	00:29,586	00:29,432	00:29,323	00:28,974

Programa A

72					
	1mW	25mW	100mW	200mW	300mW
100 m	00:32,244	00:32,250	00:31,989	00:31,782	00:31,950
500 m	00:32,501	00:32,432	00:32,227	00:32,123	00:31,792
1.000 m	00:32,849	00:32,654	00:32,132	00:32,458	00:32,123
2.000 m	00:32,981	00:32,677	00:32,345	00:32,531	00:31,792

Anexo 1

Media de tiempo de recepción completo del programa A, según tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868

Programa A	100	94	90	80	72
1mW	00:22,598	00:24,802	00:25,832	00:29,106	00:32,644
25mW	00:22,415	00:24,665	00:25,759	00:29,005	00:32,503
100mW	00:22,328	00:24,425	00:25,523	00:28,859	00:32,173
200mW	00:21,976	00:24,390	00:25,338	00:28,774	00:32,224
300mW	00:21,918	00:24,111	00:25,345	00:28,664	00:31,914

Tiempo medio de recepción y procesamiento por paquete según distancia entre nodos, tamaño de la carga útil del paquete y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
100					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,107	00:01,105	00:01,094	00:01,097	00:01,084
500 m	00:01,133	00:01,117	00:01,127	00:01,093	00:01,122
1.000 m	00:01,130	00:01,123	00:01,117	00:01,083	00:01,079
2.000 m	00:01,149	00:01,138	00:01,127	00:01,123	00:01,099

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,111	00:01,109	00:01,096	00:01,093	00:01,085
500 m	00:01,126	00:01,114	00:01,099	00:01,098	00:01,087
1.000 m	00:01,127	00:01,125	00:01,116	00:01,114	00:01,096
2.000 m	00:01,145	00:01,136	00:01,130	00:01,130	00:01,116

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,105	00:01,104	00:01,097	00:01,087	00:01,105
500 m	00:01,116	00:01,117	00:01,101	00:01,096	00:01,087
1.000 m	00:01,127	00:01,125	00:01,117	00:01,111	00:01,106
2.000 m	00:01,145	00:01,134	00:01,124	00:01,112	00:01,110

Programa A					
80					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,094	00:01,103	00:01,092	00:01,089	00:01,091
500 m	00:01,115	00:01,107	00:01,102	00:01,098	00:01,094
1.000 m	00:01,131	00:01,115	00:01,113	00:01,112	00:01,111
2.000 m	00:01,139	00:01,138	00:01,132	00:01,128	00:01,114

Programa A					
72					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,112	00:01,112	00:01,103	00:01,096	00:01,102
500 m	00:01,121	00:01,118	00:01,111	00:01,108	00:01,096
1.000 m	00:01,133	00:01,126	00:01,108	00:01,119	00:01,108
2.000 m	00:01,137	00:01,127	00:01,115	00:01,122	00:01,096

Media del tiempo medio de recepción y procesamiento por paquete, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	00:01,106	00:01,107	00:01,096	00:01,092	00:01,093
500 m	00:01,122	00:01,114	00:01,108	00:01,098	00:01,097
1.000 m	00:01,130	00:01,123	00:01,114	00:01,108	00:01,100
2.000 m	00:01,143	00:01,135	00:01,126	00:01,123	00:01,107

Anexo 1

Número de reenvíos y tasa de reenvíos, según tamaño de la carga útil del paquete, distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A

100					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	1
500 m	3	2	2	3	1
1.000 m	5	6	6	4	5
2.000 m	6	8	7	6	4

Programa A

100					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	5,00%
500 m	15,00%	10,00%	10,00%	15,00%	5,00%
1.000 m	25,00%	30,00%	30,00%	20,00%	25,00%
2.000 m	30,00%	40,00%	35,00%	30,00%	20,00%

Programa A

94					
	1mW	25mW	100mW	200mW	300mW
100 m	1	0	0	1	1
500 m	3	2	1	4	2
1.000 m	4	4	3	5	3
2.000 m	5	6	8	6	5

Programa A					
94					
	1mW	25mW	100mW	200mW	300mW
100 m	4,55%	0,00%	0,00%	4,55%	4,55%
500 m	13,64%	9,09%	4,55%	18,18%	9,09%
1.000 m	18,18%	18,18%	13,64%	22,73%	13,64%
2.000 m	22,73%	27,27%	36,36%	27,27%	22,73%

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	0	0	0	0	0
500 m	2	3	2	2	1
1.000 m	3	2	2	4	3
2.000 m	6	5	5	4	3

Programa A					
90					
	1mW	25mW	100mW	200mW	300mW
100 m	0,00%	0,00%	0,00%	0,00%	0,00%
500 m	8,70%	13,04%	8,70%	8,70%	4,35%
1.000 m	13,04%	8,70%	8,70%	17,39%	13,04%
2.000 m	26,09%	21,74%	21,74%	17,39%	13,04%

Programa A

80					
	1mW	25mW	100mW	200mW	300mW
100 m	1	0	1	1	1
500 m	3	3	4	1	2
1.000 m	4	3	3	3	4
2.000 m	7	7	3	6	8

Programa A

80					
	1mW	25mW	100mW	200mW	300mW
100 m	3,85%	0,00%	3,85%	3,85%	3,85%
500 m	11,54%	11,54%	15,38%	3,85%	7,69%
1.000 m	15,38%	11,54%	11,54%	11,54%	15,38%
2.000 m	26,92%	26,92%	11,54%	23,08%	30,77%

Programa A

72					
	1mW	25mW	100mW	200mW	300mW
100 m	1	1	0	0	1
500 m	4	5	4	3	3
1.000 m	2	3	3	3	4
2.000 m	9	6	8	9	9

Programa A					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	2,37%	0,69%	0,77%	1,68%	3,37%
500 m	12,53%	12,18%	10,48%	11,21%	7,30%
1.000 m	15,70%	15,75%	14,84%	16,40%	16,17%
2.000 m	27,35%	27,32%	26,45%	25,76%	23,51%

Tasa media de reenvíos, según distancia entre nodos y potencia del módulo XBee-PRO 868 para el programa A

Programa A					
media					
	1mW	25mW	100mW	200mW	300mW
100 m	2,37%	0,69%	0,77%	1,68%	3,37%
500 m	12,53%	12,18%	10,48%	11,21%	7,30%
1.000 m	15,70%	15,75%	14,84%	16,40%	16,17%
2.000 m	27,35%	27,32%	26,45%	25,76%	23,51%

A.3. Potencia de la señal - Programa A

Potencia de la señal(dBm) para el programa A con tamaño 72 bytes de carga útil de paquete, según distancia entre nodos y potencia del módulo XBee-PRO 868 para cada uno de los 31 paquetes enviados

100 m	1mW	25mW	100mW	200mW	300mW
1	-96	-97	-103	-94	-103
2	-97	-95	-95	-94	-95
3	-94	-93	-98	-94	-98
4	-95	-95	-92	-95	-97
5	-98	-96	-92	-100	-100
6	-95	-92	-95	-96	-95
7	-92	-94	-93	-100	-95
8	-91	-93	-95	-97	-94
9	-94	-95	-94	-101	-97
10	-96	-93	-93	-95	-94
11	-95	-94	-96	-100	-94
12	-95	-95	-94	-97	-94
13	-94	-94	-91	-99	-94
14	-95	-94	-94	-102	-94
15	-96	-94	-92	-97	-95
16	-97	-91	-96	-96	-100
17	-92	-99	-95	-100	-96
18	-99	-95	-95	-101	-92
19	-92	-94	-96	-95	-95
20	-95	-94	-94	-100	-97
21	-96	-96	-94	-96	-96
22	-96	-93	-96	-97	-97

Anexo 1

23	-92	-96	-98	-97	-95
24	-93	-96	-95	-98	-93
25	-92	-96	-95	-97	-94
26	-91	-92	-95	-97	-94
27	-91	-93	-94	-96	-95
28	-95	-92	-94	-97	-95
29	-96	-94	-93	-97	-94
30	-95	-96	-95	-96	-95
31	-94	-93	-93	-102	-92

Anexo 1

500 m	1mW	25mW	100mW	200mW	300mW
1	-94	-110	-98	-101	-99
2	-102	-100	-96	-96	-97
3	-104	-99	-98	-97	-99
4	-102	-99	-101	-104	-99
5	-99	-98	-97	-97	-100
6	-96	-95	-98	-99	-98
7	-97	-102	-100	-100	-98
8	-99	-100	-100	-98	-97
9	-102	-98	-98	-100	-96
10	-102	-97	-99	-100	-100
11	-100	-97	-95	-99	-102
12	-96	-102	-100	-96	-102
13	-99	-100	-102	-98	-96
14	-99	-100	-100	-98	-96
15	-97	-102	-97	-97	-104
16	-98	-99	-97	-97	-99
17	-103	-96	-98	-99	-100
18	-100	-100	-103	-102	-100
19	-102	-99	-98	-98	-99
20	-98	-101	-102	-102	-104
21	-98	-98	-99	-98	-101
22	-102	-98	-96	-102	-96
23	-103	-98	-101	-98	-101
24	-99	-101	-95	-97	-98
25	-100	-96	-102	-99	-101

Anexo 1

26	-104	-103	-96	-100	-97
27	-103	-100	-100	-100	-97
28	-98	-97	-96	-100	-104
29	-104	-101	-98	-98	-100
30	-98	-95	-98	-98	-102
31	-98	-99	-101	-100	-99

Anexo 1

1 km	1mW	25mW	100mW	200mW	300mW
1	-107	-99	-97	-99	-98
2	-99	-106	-96	-102	-103
3	-102	-100	-102	-100	-99
4	-106	-104	-99	-97	-102
5	-100	-100	-103	-101	-97
6	-97	-99	-99	-105	-99
7	-104	-96	-100	-99	-100
8	-105	-106	-100	-102	-96
9	-106	-100	-99	-97	-92
10	-103	-105	-100	-96	-95
11	-104	-102	-101	-100	-101
12	-101	-100	-105	-101	-97
13	-96	-103	-97	-95	-99
14	-102	-99	-99	-100	-100
15	-97	-100	-100	-96	-96
16	-101	-104	-97	-97	-102
17	-105	-99	-96	-97	-103
18	-106	-105	-102	-96	-95
19	-102	-101	-100	-106	-98
20	-107	-105	-97	-100	-97
21	-99	-106	-96	-97	-97
22	-102	-108	-106	-96	-100
23	-100	-101	-102	-98	-102
24	-104	-106	-100	-102	-99

Anexo 1

25	-105	-99	-97	-100	-102
26	-106	-100	-96	-99	-97
27	-101	-105	-106	-101	-99
28	-106	-99	-99	-100	-99
29	-102	-104	-99	-102	-100
30	-100	-97	-101	-102	-96
31	-97	-101	-103	-100	-102

Anexo 1

2 km	1mW	25mW	100mW	200mW	300mW
1	-102	-105	-102	-100	-98
2	-104	-101	-100	-97	-99
3	-104	-106	-97	-101	-102
4	-105	-101	-101	-99	-99
5	-104	-100	-100	-100	-101
6	-102	-97	-99	-96	-97
7	-105	-99	-100	-102	-100
8	-104	-106	-97	-102	-97
9	-105	-100	-96	-98	-96
10	-107	-97	-102	-103	-97
11	-102	-101	-99	-99	-98
12	-104	-105	-99	-102	-103
13	-105	-106	-102	-100	-99
14	-102	-108	-102	-97	-102
15	-102	-104	-100	-96	-97
16	-106	-100	-97	-106	-99
17	-104	-97	-96	-101	-102
18	-104	-96	-106	-100	-100
19	-103	-106	-102	-100	-97
20	-103	-108	-100	-97	-96
21	-104	-102	-97	-99	-100
22	-101	-103	-100	-101	-97
23	-103	-110	-100	-100	-99
24	-106	-104	-102	-102	-100
25	-103	-109	-97	-99	-100

Anexo 1

26	-104	-106	-98	-99	-96
27	-102	-102	-102	-101	-97
28	-103	-100	-100	-100	-100
29	-103	-97	-106	-101	-102
30	-102	-100	-102	-95	-98
31	-106	-97	-100	-99	-103

Potencia media de la señal(dBm) para el programa A, según distancia entre nodos y potencia del módulo XBee-PRO 868

dBm	1mW	25mW	100mW	200mW	300mW
100 m	-94,484	-94,323	-94,677	-97,516	-95,452
500 m	-99,871	-99,355	-98,677	-98,968	-99,387
1000 m	-102,323	-101,903	-99,806	-99,452	-98,774
2000 m	-103,677	-102,355	-100,032	-99,742	-99,065