



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

TÍTULO DEL TFG: Estudio de las prestaciones energéticas de SigFox y NB-IoT

TITULACIÓN: Grado en Ingeniería Telemática

AUTOR: Juan Carlos Veras Revelles

DIRECTOR: Carles Gómez Montenegro

FECHA: 31 de diciembre de 2018

Título: Estudio de las prestaciones energéticas de Sigfox y NB-IoT

Autor: Juan Carlos Veras Revelles

Director: Carles Gómez Montenegro

Fecha: 31 de diciembre de 2018

Resumen

Internet de las cosas (IoT) está teniendo cada vez más presencia en nuestra sociedad y la industria apuesta por su desarrollo y evolución. IoT representa una nueva forma de comunicación entre máquinas sin requerir de interacciones humanas con el fin de que diferentes dispositivos y sensores sean capaz de transmitir datos a la red acerca del mundo físico, estos puedan ser analizados e interpretados obteniendo así información útil que ayude a tomar decisiones, monitorear o almacenar datos, entre otros ejemplos.

En la mayoría de casos, estos dispositivos o sensores no se encuentran conectados a la red eléctrica y requieren de fuentes autónomas de energía para abastecerse. Esta circunstancia es relevante a la hora de poder desplegar redes que den cabida a estos dispositivos. Por ello, nuevas tecnologías han sido desarrolladas recientemente con el objetivo de permitir el correcto desarrollo y despliegue de estos dispositivos. Así surgieron las denominadas Low Power Wide Area Network. Estas tecnologías están diseñadas para poder ofrecer comunicaciones de largo alcance empleando una baja utilización de recursos energéticos. Dentro de estas tecnologías, dos ejemplos relevantes son SigFox y Narrowband-IoT.

El objetivo principal de este proyecto es evaluar el consumo energético de estas dos tecnologías utilizando diferentes plataformas disponibles en el mercado, herramientas de laboratorio como un analizador de potencia y diferente software para configurar los dispositivos y obtener resultados. De este modo, podemos generar un modelo realista de consumo energético con ambas tecnologías y, a partir de los resultados obtenidos, estimar el tiempo de vida y el coste energético de entrega de datos de los dispositivos que emplean las tecnologías mencionadas.

Title: Study of the energy performance of Sigfox and NB-IoT

Author: Juan Carlos Veras Revelles

Director: Carles Gómez Montenegro

Date: December 31 th 2018

Overview

Internet of things (IoT) is having an increasing presence in our society and the industry is committed to its development and evolution. IoT represents a new form of communication between machines without requiring human interactions in order that different devices and sensors are able to transmit data to the network about the physical world, these can be analyzed and interpreted thus obtaining useful information to help take decisions, monitor or store data among other examples.

In most cases, these devices or sensors are not connected to the power grid and require autonomous sources of energy to be powered. This circumstance is relevant when it comes to deploying networks that fits these devices. Therefore, new technologies have been developed recently in order to allow the correct development and deployment of these devices. This is how Low Power Wide Area Network came about. These technologies are designed to be able to offer long-range communications using a low use of energy resources. Within these technologies, two relevant examples are SigFox and Narrowband-IoT.

The main objective of this project is to evaluate the energy consumption of these two technologies using different platforms available in the market, laboratory tools such as a power analyzer and different software to configure the devices and obtain the results. In this way, a realistic model of energy consumption can be generated with both technologies and, based on the results obtained, estimate the lifetime and the energy cost of data delivery of devices that use these technologies.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. TECNOLOGÍAS LPWAN.....	3
1.1. Sigfox.....	3
1.1.1. Red Sigfox	3
1.1.2. Transmisión radio	5
1.1.3. Características de enlace	5
1.1.4. Aspectos de seguridad	6
1.1.5. Formato de trama física.....	6
1.1.6. Comunicación de datos	8
1.2. NB-IoT	10
1.2.1. Red NB-IoT	11
1.2.2. Transmisión radio	12
1.2.3. Características de enlace	13
1.2.4. Power-Saving Mode	13
CAPÍTULO 2. ENTORNO EXPERIMENTAL	15
2.1. Hardware Sigfox	15
2.1.1. Plataforma	15
2.1.2. Antena	17
2.2. Software Sigfox	17
2.2.1. Arduino IDE	17
2.2.2. Registro de dispositivo	18
2.2.3. Librería Sigfox	18
2.2.4. Código fuente	19
2.3. Hardware NB-IoT	21
2.3.1. Plataforma	21
2.4. Software NB-IoT.....	23
2.4.1. Comandos AT.....	23
2.4.2. Código script.....	23
2.5. Analizador de potencia	26
CAPÍTULO 3. MODELO DE ENERGÍA	28
3.1. Sigfox.....	28
3.1.1. Transmisión sin mensaje de bajada.....	29
3.1.2. Transmisión con mensaje de bajada.....	31

3.2. NB-IoT	34
3.2.1. Transmisión de mensaje	35
CAPÍTULO 4. RESULTADOS Y DISCUSIÓN	38
4.1. Evaluación de Sigfox	38
4.1.1. Consumo de corriente	38
4.1.2. Tiempo de vida	42
4.1.3. Coste energético	44
4.2. Evaluación de NB-IoT	47
4.2.1. Consumo de corriente	47
4.2.2. Tiempo de vida	48
4.2.3. Coste energético	48
CAPÍTULO 5. CONCLUSIONES	50
5.1. Conclusiones	50
5.2. Líneas futuras	51
BIBLIOGRAFÍA	52

INTRODUCCIÓN

El Internet de las cosas (IoT) es un término cada vez más presente en la sociedad. Si bien es cierto, aún no se tiene bien clara la definición del concepto. Generalmente, podríamos definir el IoT como la conexión a Internet de objetos físicos interconectados, principalmente utilizando tecnología inalámbrica aunque no necesariamente, que incluye objetos de la vida cotidiana que pueden estar equipados con sensores y/o actuadores.

El objetivo de IoT es aprovechar al máximo el intercambio de datos entre los diferentes objetos o “cosas” y así desarrollar sistemas inteligentes que ofrezcan mejoras en la gestión de recursos, produciendo en consecuencia beneficios económicos y un impacto positivo sobre la sociedad.

Bajo este marco se creó la necesidad de diseñar nuevas tecnologías que permitiesen el correcto desarrollo y despliegue de los dispositivos IoT. Así surgieron las denominadas Low Power Wide Area Network (LPWAN).

Este tipo de tecnologías inalámbricas se presentan como gran alternativa para aprovechar su uso dentro del IoT. Están diseñadas para ofrecer comunicaciones de largo alcance utilizando una potencia baja. De esta manera los dispositivos pueden operar con baterías pequeñas durante años y en un rango generalmente superior a 2 kilómetros en entornos urbanos.

Para conseguir estas características, la mayoría de tecnologías LPWAN solo pueden enviar menos de 1000 bytes al día. Con todo esto, las LPWAN se presentan como una excelente elección donde los dispositivos necesiten enviar pocos datos sobre un área amplia y la autonomía de estos deba de ser considerable en términos de años.

Entre las diferentes tecnologías LPWAN, encontramos un estándar publicado recientemente: NarrowBand IoT (NB-IoT). Definido por el grupo de asociaciones de telecomunicaciones 3GPP en 2016 para permitir una amplia gama de dispositivos y servicios celulares, NB-IoT se centra específicamente en ofrecer gran cobertura interior, larga duración de batería, dispositivos de bajo coste y alta densidad de conexiones. Actualmente no hay muchos informes acerca de esta tecnología, y más concretamente sobre su consumo energético o rendimiento, dado que se encuentra aún en fase temprana de despliegue y expansión.

Por otro lado, otra tecnología LPWAN relevante es la definida por una empresa francesa llamada Sigfox, que proporciona un servicio de red LPWAN. Sigfox fue creado con el objetivo de ser utilizado por dispositivos de bajo consumo siendo ideal para transmisiones de datos reducidos en periodos relativamente largos.

El objetivo principal de este proyecto es llevar a cabo un estudio sobre el consumo energético de ambas tecnologías para poder estimar el tiempo de

vida y el coste energético de entrega de datos para los dispositivos que emplean las tecnologías NB-IoT y Sigfox.

El documento presenta la siguiente organización. En el primer capítulo se introducen las tecnologías Sigfox y NB-IoT, donde se detallarán este tipo de LPWAN, así como sus principales características clave para poder comprender su funcionamiento. En el segundo capítulo se describe el entorno experimental llevado a cabo para la realización de las pruebas. Se detallará las herramientas de laboratorio utilizadas, las plataformas y la configuración de las mismas. El tercer capítulo describirá los modelos de energía obtenidos para cada una de las tecnologías. El cuarto capítulo se dedica a mostrar los resultados del estudio. Se estudiará el consumo de corriente y tiempo de vida de los dispositivos, así como el coste energético de entrega de datos con las tecnologías consideradas en el proyecto. Para terminar, el quinto capítulo añadirá las conclusiones más destacadas valorando el trabajo global en conjunto y las posibles líneas futuras de trabajo.

CAPÍTULO 1. Tecnologías LPWAN

Este capítulo introducirá las tecnologías Sigfox y NB-IoT dentro del marco de las LPWAN. Se presentarán sus principales características con el fin de tener un conocimiento sobre su funcionamiento y así poder establecer una base para el análisis y tareas desarrolladas que se presentarán en los siguientes capítulos.

1.1. Sigfox

La tecnología Sigfox fue desarrollada en el año 2010 por la empresa francesa Sigfox, que a su vez es compañía y operador de red LPWAN. Dicha compañía actualmente opera en 47 países dando cobertura a más de 1.000 millones de personas y aún sigue bajo despliegue en todo el mundo gracias a la asociación con otros operadores y compañías de diversos sectores [1].

Las características principales que anuncia Sigfox y que cumplen con lo esperado de una tecnología LPWAN son:

- Bajo consumo.
- Bajo coste.
- Largo alcance.
- Mensajes cortos.

En los siguientes subapartados se detallan distintos aspectos clave sobre esta tecnología.

1.1.1. Red Sigfox

Sigfox despliega toda una infraestructura propietaria. La arquitectura de red de Sigfox consta de una única red central basada en la nube, que permite la conectividad global con un impacto mínimo en el dispositivo final y la red de acceso radio.

Aparte de los dispositivos finales, esta se compone de estaciones base equipadas con radios cognitivas definidas por software, conectadas a servidores back-end mediante red basada en IP. La topología más utilizada es la estrella de estrellas.

Los servidores back-end, como elementos centrales de la red, se encargan de la conectividad de datos entre las estaciones base e Internet, así como del control y la gestión de las estaciones base y dispositivos finales. Una gran ventaja de esta red es que mediante HTTP callbacks fácilmente los datos

recibidos de los dispositivos finales pueden ser reenviados hacia otros servidores de cliente con el fin de gestionar aplicaciones IoT.

La red de acceso radio consta de varias estaciones base conectadas directamente a servidores back-end, donde cada estación base realiza funciones complejas L1 / L2, dejando algunas funcionalidades L2 y L3 a los servidores. Los dispositivos finales son los objetos, como sensores o actuadores, que comunican los datos de la aplicación IoT.

Cabe destacar que la comunicación entre los dispositivos finales y las estaciones base puede realizarse de manera bidireccional, aunque lo habitual es que solo se realicen envíos uplink. La comunicación downlink, es decir, desde las estaciones base hacia los dispositivos finales, solo puede ocurrir después de una comunicación uplink.

En Sigfox, el número de mensajes de subida está limitado a 140 mensajes por día con un *payload* máximo de 12 bytes, mientras que el número de mensajes de bajada está limitado a 4 mensajes por día con un *payload* de 8 bytes. Estas restricciones pueden variar ligeramente según las condiciones en las que se encuentre la red.

La siguiente figura representa la estructura básica de una red Sigfox.

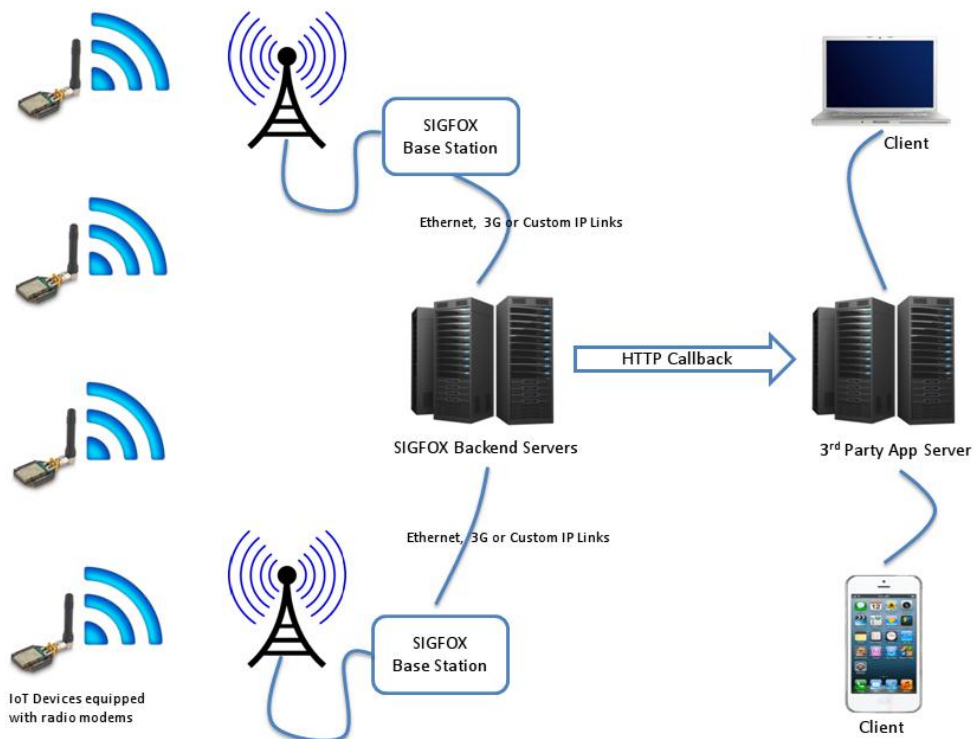


Fig. 1.1 Arquitectura de una red Sigfox [2]

Sigfox funciona bajo el modelo one-contract one-network, permitiendo a un dispositivo poder conectarse en cualquier país, prescindiendo de la idea de

roaming. Estos tampoco se asocian a una estación base determinada, sino que la información que reciben los servidores back-end puede llegar de una o varias estaciones base.

1.1.2. Transmisión radio

Para la transmisión de los mensajes entre los dispositivos finales y las estaciones base, Sigfox utiliza la modulación Differential Binary Phase-Shift Keying (DBPSK). Usar esta modulación donde la potencia de todos los símbolos es la misma supone una gran ventaja que simplifica el diseño de los elementos radio y permite un bajo coste.

Sigfox opera en bandas de frecuencia ISM (Industrial, Scientific, Medical) sin licencia, y se basa en comunicaciones UNB (Ultra Narrow Band). Esto supone una serie de ventajas:

- Bajo consumo: uso eficiente del ancho de banda de frecuencia, bajos niveles de ruido.
- Largo alcance: permite un mayor rango de transmisión mientras se consume una cantidad limitada de energía en el dispositivo.
- Escalabilidad: permite una gran cantidad de dispositivos coexistiendo en una celda determinada sin aumentar significativamente la interferencia del espectro.

Dependiendo de la región en la que se opere, se utilizará una banda de frecuencias determinada [3]:

- Europa: se utiliza la banda de frecuencia ISM de entre 868 MHz y 868.2 MHz.
- US: en Norteamérica se utiliza la banda de entre 915 MHz y 915.2 MHz.
- Asia: en esta región se utiliza la banda de entre 433 MHz y 433.2 MHz.

1.1.3. Características de enlace

Si bien la red está optimizada para comunicaciones uplink, como se ha comentado anteriormente, Sigfox soporta la comunicación tanto uplink como downlink. Debido a optimizaciones de espectro, y a las distintas restricciones en la potencia de transmisión, ambos tipos de enlace disponen de diferentes características, como se detalla a continuación.

Enlace uplink:

- Ancho de banda: 100 Hz / 600 Hz (dependiendo de la región).
- Tasa máxima de baudios: 100 baudios / 600 baudios (dependiendo de la región).

- Esquema de modulación: DBPSK.
- Potencia de transmisión: en cumplimiento con la normativa local (en Europa: 25 mW máximo con ciclo de trabajo del 1%).
- Link Budget: 155 dB o más.

Enlace downlink:

- Ancho de banda: 1.5 kHz.
- Tasa máxima de baudios: 600 baudios.
- Esquema de modulación: GFSK.
- Potencia de transmisión: 500 mW / 4 W (dependiendo de la región; en Europa: 500 mW máximo con ciclo de trabajo del 10%).
- Link Budget: 153 dB o más.

1.1.4. Aspectos de seguridad

Sigfox emplea diferentes mecanismos para poder ofrecer una red fiable y segura. El protocolo de radio autentica y asegura la integridad de cada mensaje utilizando un ID de dispositivo único y un código de autenticación de mensaje basado en AES-128. De esta manera, se asegura que el mensaje solo haya podido ser generado y enviado por el dispositivo con el ID reflejado en el mensaje.

Los datos de la aplicación se pueden cifrar a nivel de aplicación o no, dependiendo de la necesidad en cada caso de uso. Para el cifrado se utiliza AES-128 en modo contador, donde las claves criptográficas son independientes para cada dispositivo ya que están asociadas con el ID del dispositivo. Las claves de integridad y confidencialidad separadas son suministradas previamente.

1.1.5. Formato de trama física

En Sigfox se definen diferentes formatos de trama física. Se contempla un formato para las transmisiones uplink y otro para las transmisiones downlink. A continuación, se especifican los campos para cada tipo de formato.

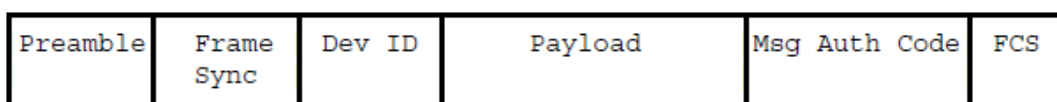


Fig. 1.2 Estructura trama física uplink.

La trama física uplink está compuesta de los siguientes campos:

- Preamble: 19 bits
- Frame sync y header: 29 bits
- Device ID: 32 bits
- Payload: 0-96 bits
- Authentication: 16-40 bits
- Frame check sequence: 16 bits (Cyclic Redundancy Check (CRC)).

Así pues, una transmisión enviada desde un dispositivo final hacia una estación base podrá tener una longitud mínima de 14 bytes y una longitud máxima de 29 bytes.

Los primeros bits de la trama (Preamble y Frame sync) se utilizan con fines de sincronización entre los dispositivos emisores y los receptores. Permite que un receptor sincronice el reloj para encontrar correctamente el comienzo de cada uno de los bits.

El campo Device ID es el identificador de dispositivo único Sigfox. A continuación, viene el Payload que contiene los datos transmitidos por el dispositivo, seguido del Msg Auth Code que sirve para poder autenticar el mensaje. Por último, el campo FCS se utiliza para comprobar que el mensaje mantiene su integridad.

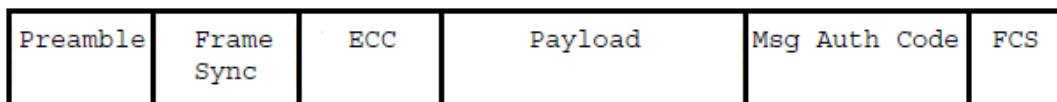


Fig. 1.3 Estructura trama física downlink.

Para el caso de las transmisiones downlink, la trama está compuesta por los siguientes campos:

- Preamble: 91 bits
- Frame sync y header: 13 bits
- ECC: 32 bits
- Payload: 0-64 bits
- Authentication: 16 bits
- Frame check sequence: 8 bits (Cyclic Redundancy Check (CRC)).

En este caso, una transmisión enviada desde una estación base hacia un dispositivo final podrá tener una longitud mínima de 20 bytes y una longitud máxima de 28 bytes. La única diferencia con respecto al caso anterior la encontramos en el campo ECC (Error Correcting Code), utilizado para corrección de errores.

1.1.6. Comunicación de datos

En Sigfox, los mensajes uplink son iniciados por el dispositivo final y son retransmitidos por una o varias estaciones base hacia los servidores back-end. Por otro lado, los mensajes downlink son aquellos enviados por el servidor back-end de la red hacia un único dispositivo final a través de una única estación base.

Como se ha comentado al comienzo de este capítulo, existe la posibilidad de solicitar un mensaje downlink que puede servir entre otros usos como comprobación de que el mensaje transmitido por el dispositivo ha sido recibido correctamente.

Un punto fuerte de la calidad de servicio de Sigfox reside en que la transmisión de un mensaje se realiza bajo lo que se conoce como diversidad en tiempo y frecuencia.

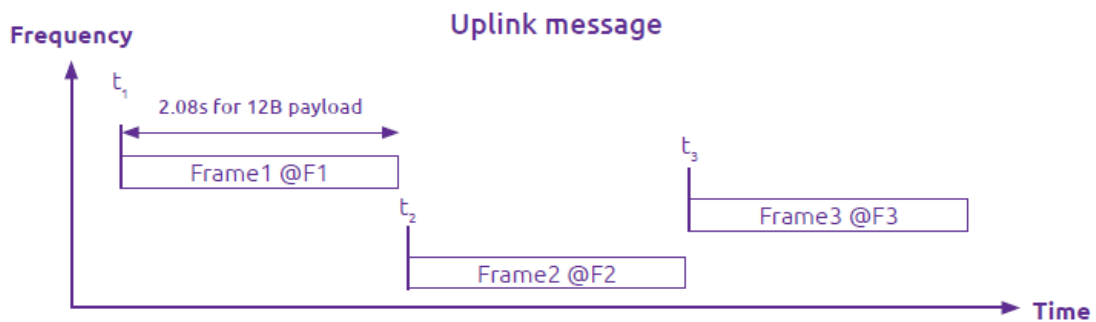


Fig. 1.4 Diversidad en tiempo y frecuencia [4]

Un mensaje es transmitido en un canal A seguido de dos réplicas transmitidas aleatoriamente en un canal B y otro canal C diferentes al primero. De esta manera se garantiza en gran medida la entrega del mensaje, evitando así posibles interferencias en un canal determinado, por ejemplo.

Este hecho, así como que los mensajes pueden ser recibidos a la vez por varias estaciones base, representan los factores clave que hay detrás de la calidad de servicio en la red Sigfox.

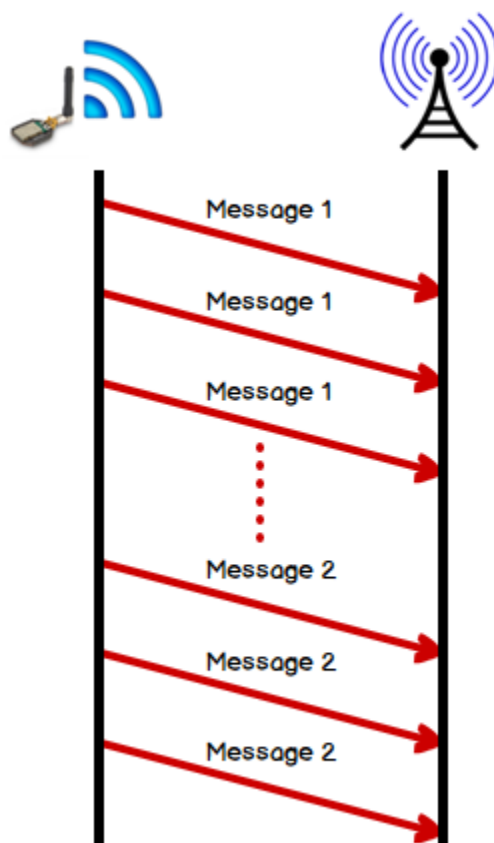


Fig. 1.5 Flujo de una transmisión sin mensaje downlink.

Por otro lado, un dispositivo puede requerir de un mensaje downlink para, por ejemplo, realizar una acción, administrar un dispositivo o configurar los parámetros de la aplicación de forma remota.

En este caso, un dispositivo dispuesto a recibir mensajes downlink abre una ventana fija para la recepción después de enviar la última réplica del mensaje uplink. El retraso y la duración de esta ventana tienen valores fijos. Existe un retraso de 20 segundos entre el primer mensaje transmitido y la ventana de recepción que dura un máximo de 25 segundos. La frecuencia del enlace downlink es la frecuencia del primer mensaje de enlace uplink más una delta conocida.

La red transmite el mensaje downlink para un dispositivo dado durante la ventana de recepción. También se selecciona la estación base para transmitir dicho mensaje. Las transmisiones de enlace uplink y downlink están desequilibradas debido a las restricciones regulatorias en las bandas ISM.

Por último, una vez el dispositivo ha recibido el mensaje downlink, este transmite un mensaje único ACK uplink terminando así el flujo de transmisión.

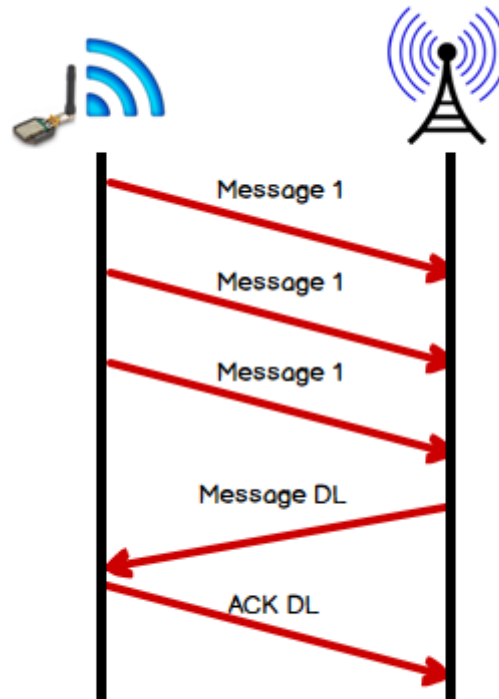


Fig. 1.6 Flujo de una transmisión con mensaje downlink.

1.2. NB-IoT

Narrowband Internet of Things es una tecnología LPWAN desarrollada y estandarizada por 3rd Generation Partnership Project (3GPP). Se basa en el uso de tecnología radio de banda estrecha. Sus especificaciones se publicaron en la versión 13 del 3GPP en junio de 2016.

En diciembre de ese año, Vodafone y Huawei integraron NB-IoT en la red española de Vodafone y enviaron el primer mensaje, conforme al estándar, hacia un dispositivo instalado en un medidor de agua. No obstante, esta tecnología todavía se encuentra en fase de pruebas en Europa por lo que aún no está muy presente en el mercado. Mejoras adicionales se han especificado en la versión 14 3GPP de 2017, como por ejemplo, el soporte a multicast.

En mayo de 2017, el gobierno de China anunció la decisión de acelerar el uso comercial de NB-IoT para desarrollar aplicaciones de Smart Cities y otros servicios públicos. Actualmente, Huawei está multiplicando las asociaciones para implementar esta tecnología durante el 2018 en varios países.

1.2.1. Red NB-IoT

NB-IoT, aunque está integrado en la norma LTE, puede considerarse como una nueva interfaz radio desde el punto de vista de la pila de protocolos [5]. Con el objetivo de reducir los costos de dispositivo y mantenimiento, algunas características LTE no están incluidas como el traspaso, mediciones para monitorear la calidad del canal, agregación de portadoras o conectividad dual.

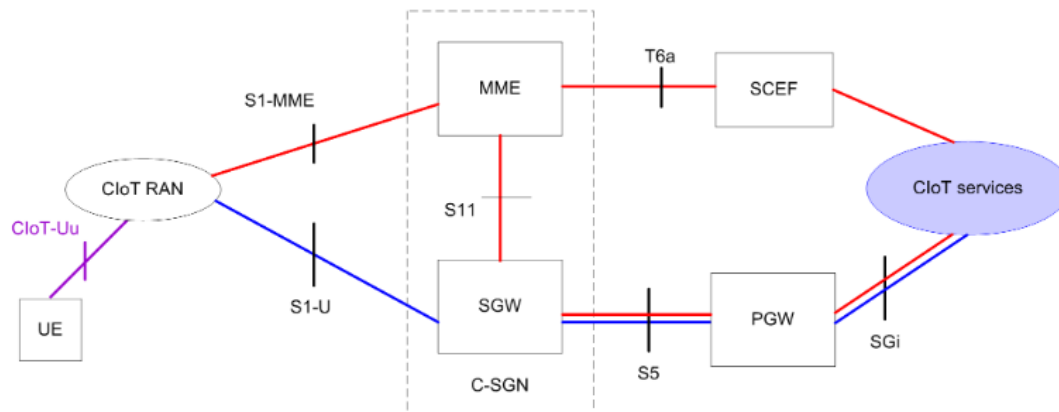


Fig. 1.7 Arquitectura de red NB-IoT [6].

La red central se basa en EPS (Evolved Packet System) junto con dos optimizaciones que se definieron para el CloT (Cellular Internet of Things): la optimización de CloT EPS en el plano de usuario (línea azul, Fig. 1.7) y la optimización de CloT EPS en el plano de control (línea roja, Fig. 1.7). Ambos planos eligen la mejor ruta para el control y los paquetes de datos de usuario, los datos de enlace ascendente y de enlace descendente. En la figura anterior se muestran los elementos que participan en la red:

- MME: Mobility Management Entity
- SGW: Serving Gateway
- PGW: Packet Data Network Gateway
- SCEF: Service Capability Exposure Function
- RAN: Radio Access Network
- UE: User Equipment

En la optimización de CloT EPS del plano de control, los datos uplink se transfieren desde el CloT RAN hacia el MME. Desde ese punto, los datos pueden transferirse a través del SGW al PGW, o al SCEF (únicamente datos que no sean IP). Finalmente, desde estos nodos, los datos se envían al servidor de aplicaciones (CloT services). Los datos downlink se transmiten a través de los mismos caminos en la dirección inversa.

El SCEF es un nuevo nodo que se utiliza para entregar datos no IP sobre el plano de control y proporciona una interfaz para servicios de red como autenticación y autorización, descubrimiento y acceso a las capacidades de la red).

Con la optimización del Clot EPS del plano de usuario, los datos se transfieren de la misma manera que el tráfico de datos convencional, es decir, a través de los portadores de radio a través de SGW y PGW hacia el servidor de aplicaciones. Por lo tanto, genera cierta sobrecarga en la creación de la conexión, aunque, facilita una secuencia de paquetes de datos para enviar. Esta ruta es compatible con la entrega de datos tanto IP como no IP.

1.2.2. Transmisión radio

NB-IoT opera bajo bandas de frecuencias bajo licencia, coexistiendo con GSM y LTE. Dado que opera en un espectro con licencia, no tiene restricciones de acceso al canal, lo que permite un ciclo de trabajo de hasta el 100%. Para las bandas de frecuencia, se utilizan las mismas frecuencias que en LTE, con un subconjunto definido para NB-IoT. La siguiente figura muestra las frecuencias especificadas en la versión 13.

Band Number	Uplink frequency range / MHz	Downlink frequency range / MHz
1	1920 - 1980	2110 - 2170
2	1850 - 1910	1930 - 1990
3	1710 - 1785	1805 - 1880
5	824 - 849	869 - 894
8	880 - 915	925 - 960
12	699 - 716	729 - 746
13	777 - 787	746 - 756
17	704 - 716	734 - 746
18	815 - 830	860 - 875
19	830 - 845	875 - 890
20	832 - 862	791 - 821
26	814 - 849	859 - 894
28	703 - 748	758 - 803
66	1710 - 1780	2110 - 2200

Fig. 1.8 Tabla bandas de frecuencia NB-IoT [6].

Las transmisiones NB-IoT tienen un ancho de banda de 180 kHz, que corresponde a un bloque de recursos en la transmisión LTE. De esta manera se diferencian 3 modos de operación como se muestra en la siguiente figura.

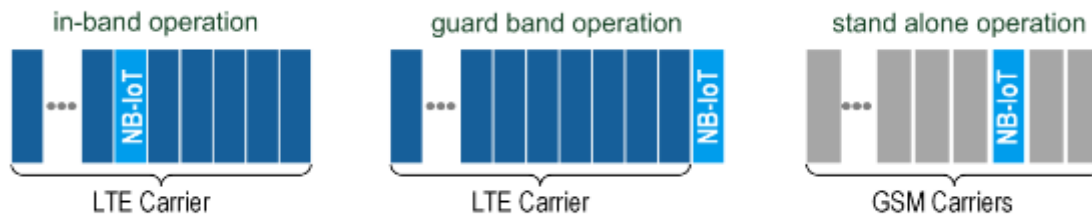


Fig. 1.9 Modos de operación NB-IoT [3].

- Modo In-band: utilizando bloques de recursos dentro de una portadora LTE. La asignación de recursos entre LTE y NB-IoT no es fija.
- Modo Guard band: utilizando bloques de recursos no utilizados dentro de la banda de guarda de una portadora LTE.
- Modo Stand alone: un posible escenario es la utilización de las bandas de frecuencias GSM actualmente utilizadas. Con ancho de banda de 200 kHz todavía resta un intervalo de guarda de 10 kHz permaneciendo en ambos lados del espectro.

NB-IoT permite la conectividad de hasta 100.000 dispositivos finales por celda con el potencial de aumentar la capacidad al agregar más portadoras. Utiliza el acceso múltiple por división de frecuencia de una sola portadora (FDMA) para enlaces uplink y FDMA ortogonal (OFDMA) para enlaces downlink.

1.2.3. Características de enlace

Este apartado recoge algunas de las características clave de los enlaces uplink y downlink [7]:

- Ancho de banda señal: 180 kHz.
- Esquema de modulación: QPSK.
- Tasa máxima de transferencia: 200 kbps (downlink) / 20 kbps (uplink).
- MTU: 1600 bytes.
- Bidireccional: half-duplex.
- Potencia de transmisión: hasta 46 dBm (downlink) / 20 – 23 dBm (uplink)
- Link Budget: 150 dB.

1.2.4. Power-Saving Mode

Para prolongar la vida útil de la batería de los dispositivos NB-IoT, se establece un modo de ahorro de energía abreviado como PSM (Power-Saving Mode). En PSM, el dispositivo está en un estado deep sleep y solo se reactiva para las transmisiones de enlace uplink.

Después de una transmisión o notificación TAU (Tracking Area Update), hay una ventana, configurada por la red, durante la cual el receptor del dispositivo está abierto para la conectividad del enlace descendente o para la señalización periódica (keep-alive). PSM utiliza la señalización TAU periódica con ventanas de recepción adicionales para la accesibilidad del enlace downlink.

El dispositivo que se encuentra en este estado consume alrededor de $5 \mu\text{A}$. La siguiente figura un ejemplo de consumo de energía de un dispositivo.

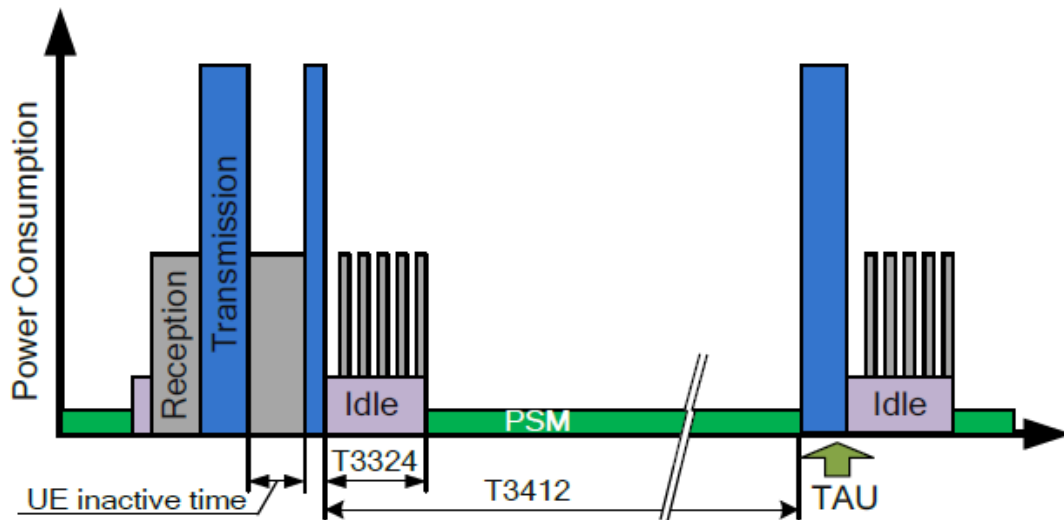


Fig. 1.10 Diagrama de consumo de energía [8].

El procedimiento establecido para entrar en el modo de ahorro de energía es el siguiente: un dispositivo solicita ingresar en el modo PSM durante el procedimiento de conexión / TAU (Actualización del área de seguimiento). La red acepta la solicitud y proporciona un valor de tiempo activo ($T3324$) al dispositivo y se inicia el temporizador móvil accesible. Cuando el temporizador $T3324$ expira, el dispositivo pasa al modo PSM durante la duración de $T3412$ (temporizador TAU periódico). Cuando se envían datos originados en el dispositivo o el temporizador TAU vence, el dispositivo sale del modo de ahorro de energía.

CAPÍTULO 2. Entorno experimental

En este capítulo se describirán todos los elementos utilizados para el desarrollo de las pruebas propósito de este trabajo. Tanto para las tecnologías Sigfox y NB-IoT se presentarán las plataformas hardware, así como sus configuraciones a nivel de software.

2.1. Hardware Sigfox

Para realizar las pruebas necesarias sobre esta tecnología, se ha optado por utilizar los elementos hardware (plataforma y antena) descritos a continuación.

2.1.1. Plataforma

Arduino MKRFOX 1200 ha sido la placa de desarrollo hardware utilizada. Esta plataforma integra un microcontrolador y un módulo con RF apto para comunicarse con la red Sigfox.

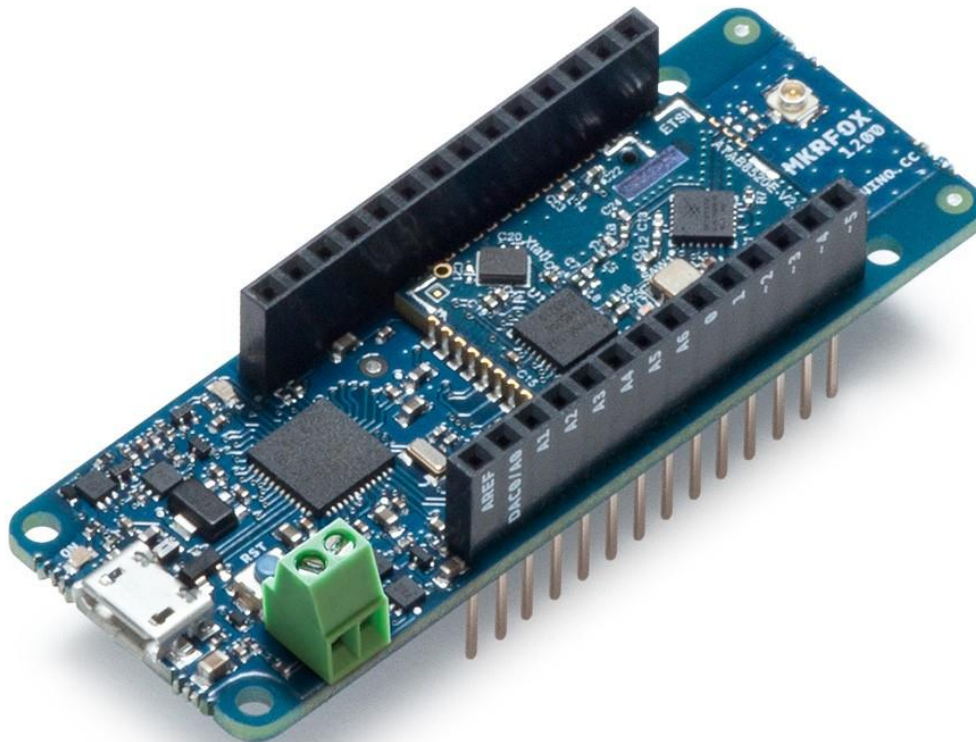


Fig. 2.1 Placa Arduino MKRFOX 1200 [9].

Como microcontrolador, la plataforma MKRFOX1200 utiliza el Atmel SAMD21 de 32 bits y el módulo Atmel ATA8520 compatible con Sigfox. A continuación, se detallan las principales características de la plataforma y estos componentes.

Tabla 2.1. Características Arduino MKRFOX1200 [10][11].

Característica	Valor
Microcontrolador	SAMD21 Cortex-M0+ 32-bit low power ARM MCU
Alimentación placa (USB/VIN)	5 V
Baterías soportadas	2x AA o AAA
Voltaje de funcionamiento	3.3 V
Digital I/O Pines	8
PWM Pines	12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 – or 18 -, A4 -or 19)
UART	1
SPI	1
I2C	1
Entradas analógicas Pines	7 (ADC 8/10/12 bit)
Salidas analógicas Pines	1 (DAC 10 bit)
Interrupciones externas	8 (0, 1, 4, 5, 6, 7, 8, A1 -or 16-, A2 – or 17)
Corriente DC para I/O Pin	7 mA
Memoria flash	256 kB
SRAM	32 kB
EEPROM	no
Velocidad reloj	32.768 kHz (RTC), 48 MHz
LED_BUILTIN	6
Dispositivo USB Full-Speed y host integrado	
LED_BUILTIN	6
Rango frecuencias operacionales	868 MHz – 868.6 MHz
Consumo corriente y potencia de salida Tx (ATA8520)	32.7 mA – 14.5dBm
Generador CRC	32 bits
Conector Antena	UFL
Longitud	67.64 mm
Ancho	25 mm
Peso	32 gr.

Cabe destacar que, con la adquisición de esta plataforma obtenemos dos años de suscripción a la red Sigfox con la posibilidad de enviar hasta 140 mensajes por día, 1 cada 10 minutos aproximadamente, lo que supone un total máximo de 1680 bytes de datos útiles enviados por día.

La placa se alimenta mediante USB a 5 V. Sólo es recomendable alimentar así a la placa cuando se programa. La placa está diseñada para alimentarse a 3 V a través del bornero, en este caso se podría utilizar un portapilas y utilizar dos pilas AA o AAA.

2.1.2. Antena

Para realizar las transmisiones, se ha utilizado la antena suministrada junto a la plataforma. Dicha antena también es compatible con GSM (850 / 900 / 1800 / 1900 MHz) y es apta para usar con la placa de desarrollo MKRFOX1200. Se conecta a la placa de desarrollo a través de un conector Micro U.FL. Tiene una ganancia de 2 dB y un tamaño reducido (130 mm x 16 mm x 5 mm), no tiene plomo y cumple con la directiva RoHS.



Fig. 2.2 Antena compatible con la banda 868 MHz [9].

2.2. Software Sigfox

En esta sección se describen las herramientas software utilizadas y el desarrollo de código necesario para los experimentados realizados con la tecnología Sigfox.

2.2.1. Arduino IDE

Para trabajar con la plataforma y programarla, se ha utilizado el entorno de programación Arduino IDE. En primer lugar, tenemos que instalar las siguientes librerías y el driver de la placa:

- Arduino SAMD Boards (32-bits ARM Cortex-M0+)
- Arduino Sigfox for MKRFox1200
- Arduino Low Power
- RTCZero

El entorno Arduino IDE permite gestionar la conexión con el dispositivo y poder editar código escrito en C, para después flashearlos en la memoria del dispositivo y correr el programa.

2.2.2. Registro de dispositivo

Antes de poder enviar mensajes a la red Sigfox, debemos registrar el dispositivo en la plataforma. Cada dispositivo que se conecta a Sigfox tiene un identificador único llamado ID que identifica a dicho dispositivo en la red Sigfox de forma única.

Por otro lado también tenemos que proporcionar el PAC (Porting Authorization Code). Este código se asocia al propietario del dispositivo y sólo él lo conoce. Sirve para que nadie, salvo el propietario que conoce el código PAC, pueda añadir ese dispositivo a un grupo y tomar el control del mismo. Estos dos parámetros se extraen de la propia memoria del módulo de ATA8520.

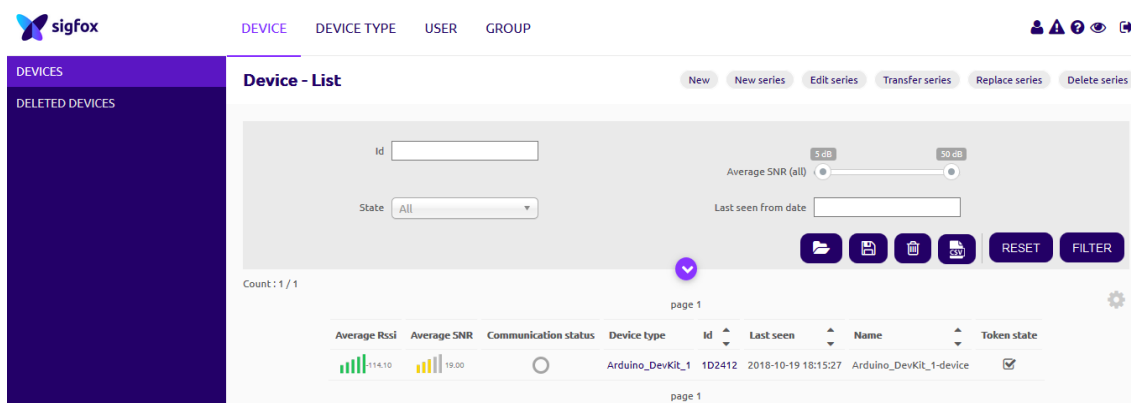


Fig. 2.3 Pantalla de registro del backend Sigfox.

2.2.3. Librería Sigfox

Para poder desarrollar los diferentes códigos a cargar en el dispositivo, se ha utilizado la librería Arduino Sigfox. En ella se definen diferentes funciones para poder realizar comunicaciones con la red. La siguiente tabla muestra algunas de las funciones más relevantes y utilizadas para este proyecto.

Tabla 2.2. Funciones librería Arduino Sigfox [12].

Función	Descripción
begin()	Inicializa el módulo y la librería.
beginPacket()	Comienza el proceso de mandar un paquete.
write()	Manda datos binarios al backend de Sigfox.

print()	Manda caracteres al backend de Sigfox.
end()	Cierra el modulo y la librería.
endPacket()	Finaliza el proceso de enviar paquetes iniciado con beginPacket().
ID()	Devuelve el Sigfox ID del módulo que es único.
PAC()	Devuelve el PAC del módulo, que es la clave secreta correspondiente al ID. El PAC no es transferible y debe regenerarse al cambiar de dueño el módulo.
reset()	Resetea el módulo.
debug()	Habilita el debug y deshabilita las funciones de ahorro de energía.
noDebug()	Deshabilita el debug y habilita las funciones de ahorro de energía.
read()	Lee los datos entrantes de Sigfox.

2.2.4. Código fuente

Para llevar a cabo las pruebas de este proyecto, se han implementado diferentes programas. Tales programas permiten transmitir diferentes tamaños de mensaje, así como solicitar (o no) mensaje de bajada. Así pues, tenemos 4 diferentes códigos:

- Con mensaje de bajada: 1 byte y 12 bytes de payload.
- Sin mensaje de bajada: 1 byte y 12 bytes de payload.

A modo de ejemplo, a continuación se muestra el código para el caso con mensaje de bajada y 1 byte de payload.

```

#include <Arduino.h>           // required before wiring_private.h
#include <wiring_private.h>

#include <SigFox.h>
#include <ArduinoLowPower.h>

void setup() {
  Serial.begin(9600);

  if (!SigFox.begin()) { //
    return;
  }

  // Disable debug led and power saving features enabled
  SigFox.noDebug();

  delay(100);
  SigFox.end();

  for(int i = 0; i < 3; i = i+1){
    LowPower.deepSleep(10000); // Send the module to the deepest sleep
    //delay(10000);
    String message = "";
    sendString(message);
  }
}

void loop() {
}

void sendString(String str) {

  // Start the module
  SigFox.begin();
  delay(100);

  // Clears all pending interrupts
  SigFox.status();
  delay(1);

  // Send packet
  SigFox.beginPacket();
  SigFox.write(str);

  SigFox.endPacket(true); // send buffer to SIGFOX network

  SigFox.end();
}

```

Fig. 2.4 Código fuente para la transmisión con mensaje de bajada y 12 bytes de payload.

El programa anterior realiza un total de 3 transmisiones espaciadas, dejando 10 segundos entre cada una. En primer lugar, se prueba si la conexión con la red de Sigfox es correcta y se deshabilita el LED utilizado en las transmisiones

mediante la función `noDebug()`. Una vez realizada esta comprobación, y sin que haya reportado error, se procede a realizar las transmisiones.

La función `sendString()` recibe como parámetro el mensaje a enviar, arranca el módulo, abre proceso de envío de paquete y envía el mensaje string de 12 bytes de payload. Por último, cierra el proceso de envío y el módulo. Cabe destacar la función `endPacket()` pasando como parámetro un booleano `true`, que indica a la red la petición de mensaje de bajada.

Para los otros casos, la no solicitud de mensaje de bajada, se realiza pasando un booleano `false` a la función `endPacket()`. El último cambio necesario para enviar 1 byte se realiza en la función `sendString()` cambiando el tipo de variable que recibe como parámetro a tipo `byte`.

Aparte de generar estos códigos, también se ha procedido a modificar el archivo `wiring.c` de la librería `Arduino Low Power`, deshabilitando la inicialización de los pines, para así obtener el mínimo consumo durante las pruebas.

2.3. Hardware NB-IoT

Respecto a la tecnología NB-IoT, se ha optado por utilizar una única placa que reúne todos los componentes requeridos para realizar las pruebas necesarias.

2.3.1. Plataforma

Sixfab NB-IoT Evaluation Board ha sido la placa de desarrollo hardware utilizada. Esta plataforma integra un microcontrolador y una antena, reuniendo todo lo necesario para comunicarse en una única placa.

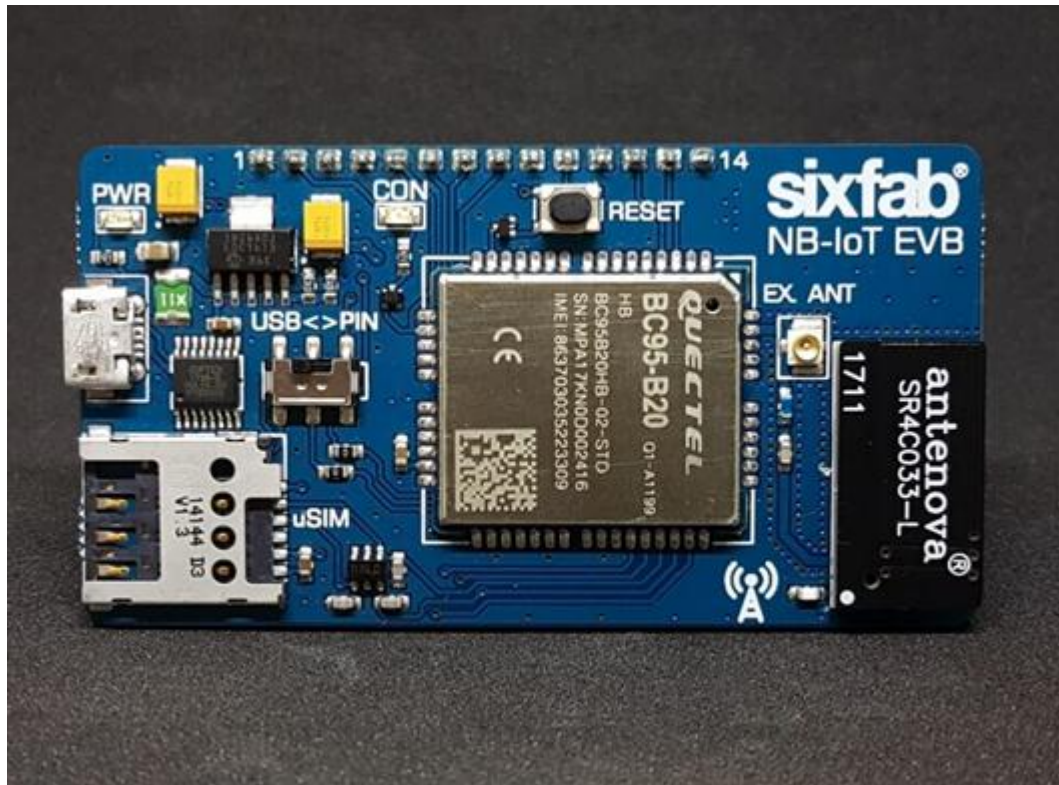


Fig. 2.5 Placa Sixfab NB-IoT Evaluation Board [13].

La plataforma Sixfab EVB posee el microcontrolador LTE BC95-B20 compatible con NB-IoT y la antena Antenova SR4C033-L, aunque con posibilidad de incorporar una antena externa mediante conector UFL. A continuación, se detallan las principales características de estos componentes.

Tabla 2.3. Características Sixfab NB-IoT Evaluation Board [14].

Característica	Valor
Microcontrolador	BC95-B20 800MHz
Alimentación placa (USB/VIN)	5V
Voltaje de funcionamiento	3.6V
Power Saving Mode	Consumo máximo 5 μ A
Potencia Tx	23 dBm \pm 2 dB
Sensibilidad	-129 dBm \pm 1dB
USIM	1
UART	2
Protocolos	IPv4 – IPv6 – UDP – CoAP
Tasa de transferencia	Single Tone 15kHz subcarrier: 24 kbps (DL), 15.625 kbps (UL)
Rango de frecuencia	791 MHz – 960 MHz
Impedancia	50 Ohms
VSWR	2.6:1
Ganancia	0.8 dBi
Comandos AT	3GPP TS 27.005, 27.007 and Quectel enhanced AT commands

2.4. Software NB-IoT

Para poder realizar las pruebas con NB-IoT es necesario configurar previamente la plataforma en la red. En los siguientes subapartados se describen los diferentes comandos utilizados y el código desarrollado.

2.4.1. Comandos AT

Los comandos AT fueron desarrollados por la compañía Hayes Communications que prácticamente se convirtieron en un estándar abierto para configurar y parametrizar módems.

Compatible con el estándar 3GPP TS 27.007 V14.3.0, la plataforma permite configurarla adecuadamente mediante comandos AT. La siguiente tabla muestra los comandos AT utilizados para este proyecto, así como otras relevantes.

Tabla 2.4. Comandos AT para configurar la plataforma [15].

Comando	Descripción
AT+CGDCONT	Definir contexto PDP (Packet Data Protocol).
AT+CFUN	Establecer funcionalidad RF del teléfono.
AT+COPS	Establecer PLMN (Public Land Mobile Network).
AT+CSQ	Obtener intensidad de señal recibida.
AT+CGATT	Establecer vinculación al packet domain service.
AT+CEREG	Estado de registro en la red.
AT+CGPADDR	Obtener dirección IP del terminal.
AT+CPSMS	Configuración Power Saving Mode.
AT+NPSMR	Estado Power Saving Mode.
AT+NRB	Reiniciar el terminal.
AT+NPING	Enviar paquete ICMP a la dirección host especificada.

2.4.2. Código script

Si bien la plataforma permite conectarse por puerto serie e ir introduciendo los comandos AT necesarios para su configuración, en este caso se ha implementado un script en Python con el objetivo de reducir al máximo el tiempo transcurrido para realizar todo el proceso de puesta en marcha y envío de paquetes ICMP satisfactoriamente.

A continuación, se muestra la parte principal del código script utilizado para configurar la plataforma en la red y realizar un envío de PING.

```
config+PING_v2.py ●
148 if __name__ == '__main__':
149
150     StartNBloT('/dev/ttyUSB0') #Port configuration
151     time.sleep(1)
152     initTime = datetime.datetime.now()
153     print "Starting time: ",initTime.time()
154     SendCommand('AT+NRB\r\n') #Reboot
155     response=get_ATresponse()
156     print "Radio status: ",response
157     time.sleep(4)
158     last = datetime.datetime.now()
159     print "Duration: +",last - initTime
160     SendCommand('AT+CGDCONT=1,"IP",""\r\n') #Set IP for PDP context
161     response=get_ATresponse()
162     print "Set IP for PDP context: ",response
163     current = datetime.datetime.now()
164     print "Duration: +",current - last
165     last = current
166     SendCommand('AT+CFUN=1\r\n') #Enable RF functions
167     response=get_ATresponse()
168     print "Radio status: ",response
169     time.sleep(2)
170     current = datetime.datetime.now()
171     print "Duration: +",current - last
172     last = current
173     SendCommand('AT+COPS=1,2,"21401"\r\n') #Set MCC and MNC values
174     response=get_ATresponse()
175     print "Set MCC and MNC: ",response
176     time.sleep(2)
177     senyal=False
```

```
178 while not senyal:
179     SendCommand('AT+CSQ\r\n') #Check the signal quality
180     response=get_ATresponse()
181     aux=get_ATresponse() # get AT OK
182     print "Quality: ",response
183     if response.find('99,99') != -1:
184         time.sleep(1)
185     else:
186         senyal=True
187         print "Received quality ok\r\n"
188     current = datetime.datetime.now()
189     print "Duration: +",current - last
190     last = current
191     SendCommand('AT+CGATT=1\r\n')
192     time.sleep(3)
193     registered=False
194     while not registered:
195         SendCommand('AT+CEREG?\r\n') #Check network register status
196         response=get_ATresponse()
197         aux=get_ATresponse() # get AT OK
198         registered=check_REG(response)
199         print "Registration status: ",response
200         time.sleep(1)
201     #####
202     print "***Test Ping 64 bytes***"
203     current = datetime.datetime.now()
204     print "Duration: +",current - last
205     last = current
206     SendCommand('AT+NPING=8.8.8.8,64\r\n') #Send Ping
207     time.sleep(1)
208     response=get_ATresponse()
209     print "PING message response: ",response
210     response=get_ATresponse()
211     print response
212     print "***Disable RF***"
213     current = datetime.datetime.now()
214     print "Duration: +",current - last
215     last = current
216     SendCommand('AT+CFUN=0\r\n') #Disable RF functions
217     response=get_ATresponse()
218     print "Radio status: ",response
219     print "Duration: +",datetime.datetime.now() - last
220     print "Total elapsed time: ",datetime.datetime.now() - initTime
```

Fig. 2.6 Código script de configuración y envío de paquete.

En el código anterior, en primer lugar, se realiza un reinicio de la plataforma para restaurar cualquier configuración previa. A continuación, se establece el protocolo IP como PDP y se habilita la funcionalidad radio para transmisión y recepción.

En el siguiente paso se define el código de red y se procede a registrar el dispositivo. También se procede a comprobar la intensidad de la señal recibida antes de continuar. Una vez comprobado esto, se asocia el PDP al dominio del servicio de paquetes y se comprueba que el dispositivo esté registrado en la red. Por último, se realiza el envío de un paquete ICMP y se deshabilitan las funcionalidades radio.

Cabe destacar que se mide el tiempo transcurrido desde el inicio hasta completar todo el proceso, así como el tiempo entre cada comando AT.

2.5. Analizador de potencia

El último elemento a presentar en este capítulo corresponde al analizador de potencia utilizado para medir el consumo de corriente de los dispositivos hardware presentados en las secciones anteriores, para ambas tecnologías. Se trata del analizador de potencia Agilent N6705A.



Fig. 2.7 Analizador de potencia Agilent N6705A [16].

Este instrumento permite alimentar los dispositivos utilizados y obtener valores de consumo de corriente, voltaje o potencia en tiempo real. Esta información se presenta en la pantalla integrada y también es posible volcarla mediante puerto USB a una memoria extraíble para después analizarla. Algunas de sus características clave son:

- 4 ranuras de fuente de alimentación DC hasta 600 W.
- Velocidad de salida de hasta 160 μ s por paso de cambio de voltaje.
- Precisión del voltímetro: hasta 0.016% + 1.5 mV, hasta 18 bits.
- Precisión del amperímetro: hasta 0.03% + 15 μ A, hasta 18 bits.

- Generador de onda arbitraria: ancho de banda de hasta 4500 Hz, salida de potencia hasta 300 W.
- Registro de datos: intervalo de medidas desde 20 μ s hasta 60 s, máximo 500 M lecturas por registro.
- 512 MB de memoria interna para guardar registros, configuraciones, etc.

Otra opción muy interesante es poder manipular de manera remota el instrumento y visualizar los datos a través del 14585A Control and Analysis Software for Advanced Power Supplies. De esta manera, podemos utilizar con mayor precisión los marcadores disponibles y realizar zoom en las zonas que nos interesen. La siguiente imagen muestra la interfaz del programa.



Fig. 2.8 Interfaz del 14585A Control and Analysis Software.

CAPÍTULO 3. Modelo de energía

En este capítulo se describirán los modelos de energía obtenidos después de realizar las medidas de corriente para ambas tecnologías, Sigfox y NB-IoT. Se mostrarán parámetros principales de rendimiento energético como el consumo de corriente, tiempo de vida de un dispositivo final o coste energético que supone el envío de un número de bytes de usuario.

Igual que en capítulos anteriores, este capítulo se dividirá en dos grandes apartados, uno para cada tecnología LPWAN considerada. Para la tecnología Sigfox, se ha tenido en cuenta los dos casos que se pueden dar en una transmisión: con y sin mensaje de bajada.

Para alimentar los dispositivos se ha utilizado el módulo de la fuente de alimentación más precisa del analizador de potencia, anteriormente presentado. A su vez, también se ha configurado el analizador de potencia para tomar muestras cada 1 ms.

3.1. Sigfox

En las pruebas realizadas se ha alimentado el dispositivo Arduino Sigfox MKRFox1200 a 3 voltios y limitado el corriente a 1 amperio. En la siguiente imagen se muestra el entorno de trabajo utilizado.

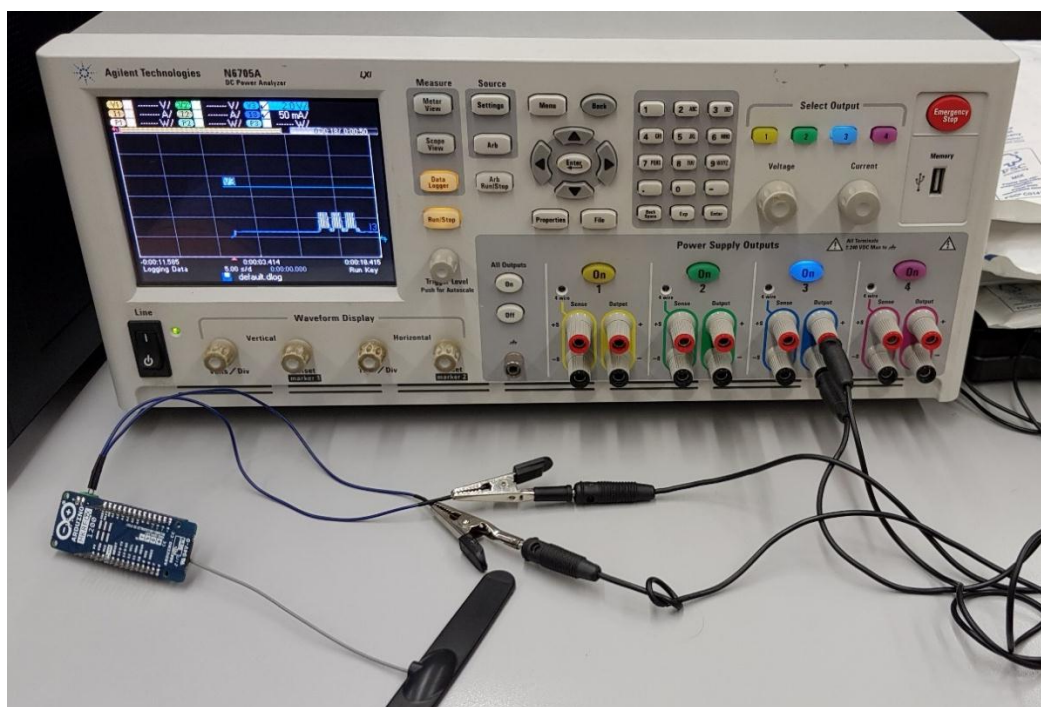


Fig. 3.1 Entorno de trabajo para la realización de las medidas con Sigfox.

3.1.1. Transmisión sin mensaje de bajada

El objetivo de este apartado es modelar el consumo de corriente medio en transmisiones que no requieren mensajes de bajada. De aquí en adelante, nos referiremos a estas transmisiones como transmisiones sin ACK.

Para calcular el consumo se tendrán en cuenta todos los estados, así como su duración y consumo medio, por los que pasa el dispositivo, desde el estado inicial (modo sleep) hasta que finaliza la transmisión y el dispositivo vuelve al estado inicial.

A continuación, se muestra el conjunto de estados que se producen al realizar transmisiones sin ACK.

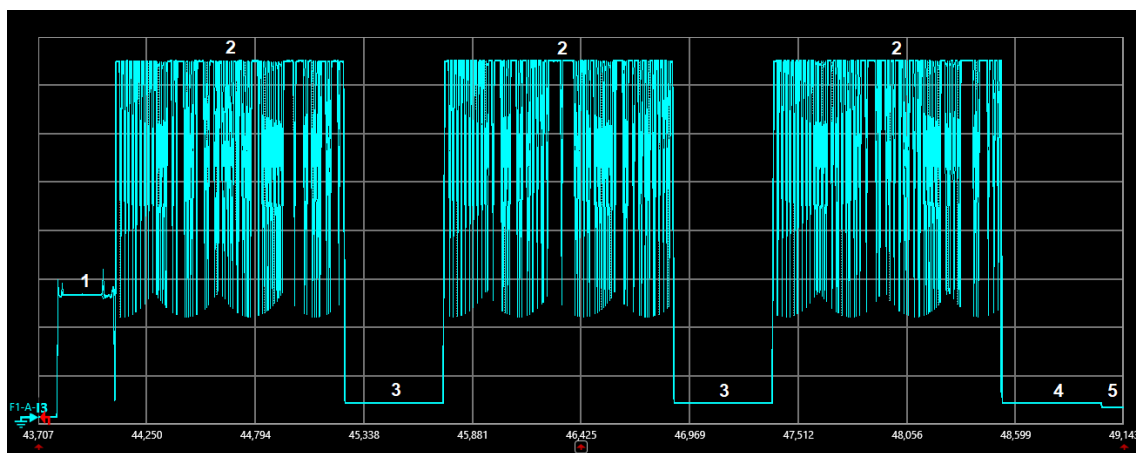


Fig. 3.2 Perfil de consumo de corriente para transmisiones sin ACK.

En la figura anterior observamos todos los estados que se dan durante la transmisión sin ACK. Inicialmente, el dispositivo permanece en un estado de reposo o sleep, destacando un consumo 2-3 órdenes de magnitud inferior al resto de estados. Cuando el dispositivo inicia el proceso de enviar mensaje, el primer estado propiamente corresponde al de wake up (estado 1), donde se prepara el mensaje para ser transmitido.

Una vez el dispositivo está listo, procede a transmitir (estado 2) el mensaje a la estación base. En la figura anterior aparece duplicado dos veces el estado 2. Estos se refieren a las retransmisiones que Sigfox realiza del mensaje utilizando diversidad en tiempo y frecuencia, tal y como se comentó en el primer capítulo.

Cabe destacar también las oscilaciones de corriente que se producen durante la transmisión. Dado que el objetivo es una transmisión Ultra Narrow Band (UNB), Sigfox aplica en la transmisión uplink power ramp-up y power ramp-down cada vez que hay un cambio de símbolo, que implica un cambio de fase, con el fin de evitar componentes de frecuencia elevados.

El estado 3 se refiere al período de espera activa que realiza el dispositivo para volver a retransmitir el mensaje. Por último, el estado 4 corresponde al momento en que el dispositivo ya ha realizado las 3 transmisiones y se dispone a entrar en el estado inicial de reposo. En la siguiente tabla se pueden observar todos los estados que se dan durante la transmisión, su duración y su consumo asociado.

Tabla 3.1. Estados y consumos asociados a una transmisión sin ACK (1 byte de payload).

#Estado	Descripción	Duración (ms)	Consumo de corriente (mA)	Variables
1	Wake up	287	10,39	T_{wu}, I_{wu}
2	Transmisión	1162	27,25	T_{tx}, I_{tx}
3	Espera Tx	486	1,24	T_{wtx}, I_{wtx}
4	Cool down	510	1,22	T_{cd}, I_{cd}
5	Sleep	Ec. (3.2)	0,016	T_{sleep}, I_{sleep}

Para calcular el consumo medio de una transmisión sin ACK en Sigfox podemos deducir la siguiente ecuación, siendo T_{notif} el tiempo entre dos transmisiones consecutivas o período, N_{states} es 5, T_i y I_i denotan el tiempo e intensidad de corriente asociados al estado i , y n_i indica el número de veces que ocurre cada estado, siendo $n_i=1$ por defecto, salvo que $n_2=3$ y $n_3=2$:

$$I_{avg_noACK} = \frac{1}{T_{notif}} \sum_{i=1}^{N_{states}} n_i \cdot I_i \cdot T_i \quad (3.1)$$

El tiempo que un dispositivo final se encuentra en el estado de más bajo consumo o sleep se puede calcular como:

$$T_{sleep} = T_{notif} - T_{act} \quad (3.2)$$

siendo T_{act} la suma los estados relacionados con la transmisión que incluyen todos menos el T_{sleep} :

$$T_{act} = T_{wu} + 3 \cdot T_{tx} + 2 \cdot T_{wtx} + T_{cd} \quad (3.3)$$

Tal y como se comentó previamente, la duración de la batería de los dispositivos finales es un factor importante a la hora de evaluar las prestaciones de la red. Una vez tenemos la intensidad media de una transmisión sin ACK, I_{avg_noACK} , se puede calcular el tiempo de vida o lifetime en

horas, $T_{lifetime_noACK}$, de una batería de capacidad $C_{battery}$ (expresada en mA h) como:

$$T_{lifetime_noACK} = \frac{C_{battery}}{I_{avg_noACK}} \quad (3.4)$$

El último parámetro de prestaciones que presentamos es el coste energético, $EC_{delivery_noACK}$. Este parámetro proporciona la energía consumida por el dispositivo final por cada bit transmitido en una transmisión no ACK. Se puede expresar como:

$$EC_{delivery_noACK} = \frac{I_{avg_noACK} \cdot V \cdot T_{notif}}{L_{delivery}} \quad (3.5)$$

siendo el numerador la energía consumida por el dispositivo durante T_{notif} . V indica el voltaje utilizado y $L_{delivery}$ indica el número de bits de información que se ha entregado de forma satisfactoria por cada trama enviada, considerando el payload de la trama únicamente.

3.1.2. Transmisión con mensaje de bajada

En este apartado se modelará el consumo de corriente medio en transmisiones con mensaje de bajada. Este modo de comunicación en la red Sigfox se da cuando el dispositivo final solicita obtener un mensaje de bajada cuando realiza la transmisión. De aquí en adelante, nos referiremos a estas transmisiones como transmisiones con ACK.

Como en el apartado anterior, para calcular el consumo se tendrán en cuenta todos los estados, así como su duración y consumo medio, por los que pasa el dispositivo, desde el estado inicial (modo sleep) hasta que finaliza la transmisión y vuelve al estado inicial.

A continuación, se muestra el conjunto de estados que se producen al realizar transmisiones con ACK.

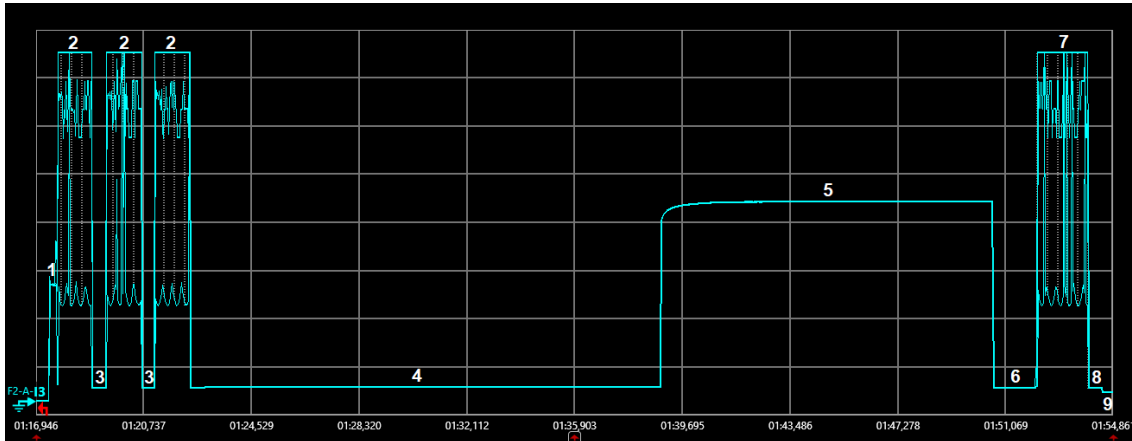


Fig. 3.3 Perfil de consumo de corriente para transmisiones con ACK.

En la figura anterior, observamos todos los estados que se dan durante la transmisión. Como en el caso anterior, inicialmente el dispositivo permanece en un estado de reposo o sleep, destacando un consumo 2-3 órdenes de magnitud inferior al resto de estados.

Cuando el dispositivo inicia el proceso de enviar mensaje, el primer estado propiamente corresponde al de wake up (estado 1), donde se prepara el mensaje para ser transmitido.

Una vez el dispositivo está listo, procede a transmitir (estado 2) el mensaje a la estación base. En este caso, también se realizan 2 transmisiones más replicando el mensaje. Después llega un período (estado 4) en que el dispositivo permanece en un estado de espera para recibir el mensaje de bajada.

Una vez finaliza dicho período, el dispositivo pasa al modo de escucha activa (estado 5) donde se espera la recepción del mensaje. La duración de este estado tiene un tiempo máximo de 25 segundos y no es fija, aunque se ha observado durante la realización de las pruebas que fluctúa muy ligeramente.

Una vez el dispositivo ha recibido el mensaje de bajada, el dispositivo procesa (estado 6) el mensaje y se prepara para transmitir un mensaje ACK con el fin de informar a la red de la recepción del mensaje de bajada.

A continuación, se transmite (estado 7) dicho mensaje ACK y por último, el dispositivo pasa a un estado de transición o cool down (estado 8) donde una vez finalizado todo el proceso de comunicación se prepara para pasar al estado inicial o modo sleep.

En la siguiente tabla se pueden observar todos los estados que se dan durante la transmisión con mensaje de bajada, su duración y consumo asociado.

Tabla 3.2. Estados y consumos asociados a una transmisión con ACK (1 byte).

#Estado	Descripción	Duración (s)	Consumo de corriente (mA)	Variables
1	Wake up	0,305	10,39	T_{wu}, I_{wu}
2	Transmisión	1,232	27,58	T_{tx}, I_{tx}
3	Espera Tx	0,493	1,23	T_{wtx}, I_{wtx}
4	Espera Rx	16,493	1,28	T_{wrx}, I_{wrx}
5	Escucha Rx	11,793	18,49	T_{lrx}, I_{lrx}
6	Preparación ACK	1,43	1,21	T_{ackp}, I_{ackp}
7	ACK Tx	1,85	26,94	T_{atx}, I_{atx}
8	Cool down	0,495	1,22	T_{cd}, I_{cd}
9	Sleep	Ec. (3.7)	0,016	T_{sleep}, I_{sleep}

Para calcular el consumo medio en este caso, procedemos de la misma manera que anteriormente, siendo T_{notif} el tiempo entre dos transmisiones consecutivas o período, N_{states} igual a 11 (3 veces se repite el estado 2 y 2 veces el estado 3) para este caso, T_i y I_i el tiempo e intensidad asociado al estado i :

$$I_{avg_ACK} = \frac{1}{T_{notif}} \sum_{i=1}^{N_{states}} n_i \cdot I_i \cdot T_i \quad (3.6)$$

El tiempo que se encuentra el dispositivo final en el estado de más bajo consumo o sleep se calcula como:

$$T_{sleep} = T_{notif} - T_{act} \quad (3.7)$$

siendo T_{act} la suma los estados relacionados con la transmisión que incluyen todos menos el T_{sleep} :

$$T_{act} = T_{wu} + 3 \cdot T_{tx} + 2 \cdot T_{wtx} + T_{wrx} + T_{lrx} + T_{ackp} + T_{atx} + T_{cd} \quad (3.8)$$

Por lo que respecta al tiempo de vida de una batería utilizando transmisión con ACK, existe cierta variación respecto al caso anterior. Se realiza una transmisión adicional (ACK confirmando mensaje de bajada) y encontramos una ventana de recepción para el mensaje de bajada. Esto comporta un aumento del consumo y por lo tanto I_{avg_ACK} (definido en (3.6)) difiere de I_{avg_noACK} (definido en (3.1)). Así pues, el tiempo de vida o lifetime en horas, $T_{lifetime_ACK}$, de una batería de capacidad $C_{battery}$ (expresada en mA h) se puede calcular como:

$$T_{lifetime_ACK} = \frac{C_{battery}}{I_{avg_ACK}} \quad (3.9)$$

Finalmente, el coste energético para una transmisión con ACK, $EC_{delivery_ACK}$ se expresa siguiendo el mismo razonamiento que en el caso anterior (ecuación (3.5)):

$$EC_{delivery_ACK} = \frac{I_{avg_ACK} \cdot V \cdot T_{notif}}{L_{delivery}} \quad (3.10)$$

siendo el numerador la energía consumida por el dispositivo durante T_{notif} . V indica el voltaje utilizado y $L_{delivery}$ indica el número de bits de información que se ha entregado de forma satisfactoria por cada trama enviada (payload).

3.2. NB-IoT

Para la realización de las medidas de consumo de corriente con NB-IoT, se ha alimentado la plataforma Sixfab NB-IoT Evaluation Board a 5 voltios y limitado el corriente a 1 amperio. En la siguiente imagen se muestra el entorno de trabajo utilizado para esta tecnología.

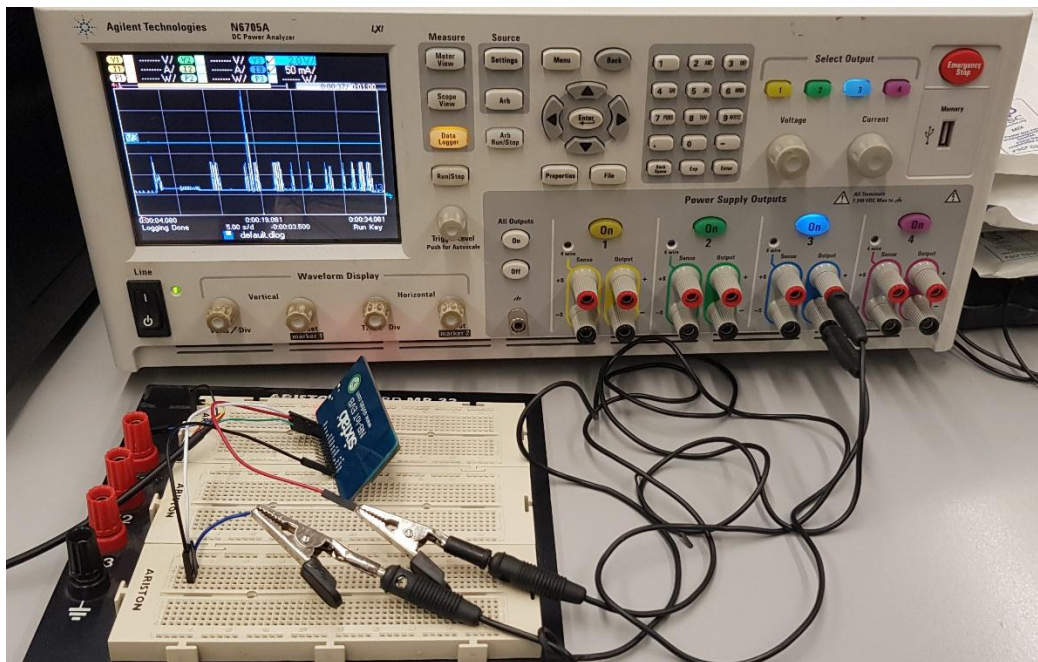


Fig. 3.4 Entorno de trabajo para la realización de las medidas con NB-IoT.

3.2.1. Transmisión de mensaje

El objetivo de este apartado es modelar el consumo de corriente medio en transmisiones NB-IoT. Para ello se ha utilizado el comando AT+NPING (tabla 2.4) para realizar transmisiones de mensajes ICMP.

Para calcular el consumo se tendrán en cuenta todos los estados observados, así como su duración y consumo medio por los que pasa el dispositivo, desde el estado inicial (modo sleep), hasta que finaliza la transmisión y vuelve al estado inicial, pasando previamente por el proceso de configuración del dispositivo.

En este caso, los estados se han establecido teniendo en cuenta el comando AT aplicado en cada instante de tiempo y su duración. A continuación, se muestra el conjunto de estados que se producen desde el reinicio del dispositivo, configuración de parámetros, registro en la red y transmisión del mensaje.

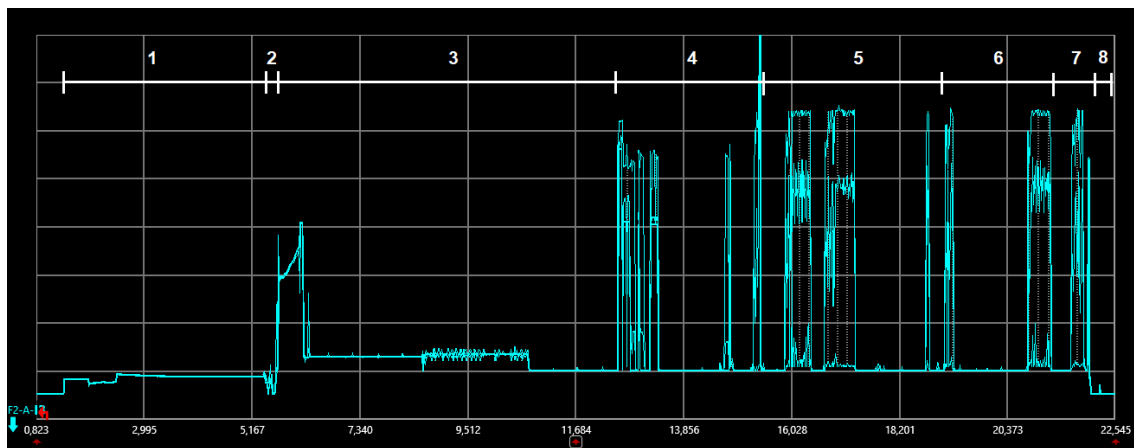


Fig. 3.5 Perfil de consumo de corriente para transmisiones de mensajes ICMP.

El primer estado corresponde al reinicio del dispositivo (estado 1) para empezar la configuración limpiamente. Esta operación se daría únicamente en la primera puesta en marcha. En el estado 2 se define el contexto PDP. El siguiente estado implica un aumento del consumo ya que se habilitan las funcionalidades RF (estado 3). Este proceso requiere de más tiempo que el resto de estados.

Una vez el dispositivo tiene la interfaz radio levantada, se procede a realizar el registro del dispositivo en la red (estado 4) y la asociación al dominio de paquete (estado 5). El estado 6 corresponde a la transmisión del paquete ICMP propiamente. Se puede observar dentro del estado, el primer pico que hace referencia al envío del ping (echo request) y el segundo pico a su respuesta (echo reply). Por último, en el estado 7 se deshabilita la interfaz radio una vez finalizada la recepción de la respuesta, y se procede a pasar al estado previo inicial de reposo.

En la siguiente tabla se pueden observar todos los estados identificados con el comando aplicado en cada uno, su duración y consumo asociado.

Tabla 3.3. Estados y consumos asociados a la configuración y transmisión de un paquete ICMP (64 bytes).

#Estado	Descripción	Duración (s)	Consumo de corriente (mA)	Variables
1	AT+NRB	4,12	10,90	T_{rb}, I_{rb}
2	AT+CGDCONT	0,13	8,36	T_{cgd}, I_{cgd}
3	AT+CFUN = 1	6,89	17,18	T_{cfun1}, I_{cfun1}
4	AT+COPS	2,27	37,21	T_{cops}, I_{cops}
5	AT+CGATT	4,19	24,99	T_{cgatt}, I_{cgatt}
6	AT+NPING	1,84	24,11	T_{ping}, I_{ping}
7	AT+CFUN = 0	0,55	39,54	T_{cfun0}, I_{cfun0}
8	Sleep	Ec. (3.12)	0,005	T_{sleep}, I_{sleep}

Para calcular el consumo medio podemos considerar la misma operativa utilizada para el caso de Sigfox. Siendo T_{notif} el período (configuración y transmisión), N_{states} igual a 7 para este caso, T_i y I_i el tiempo e intensidad asociado al estado i :

$$I_{avg} = \frac{1}{T_{notif}} \sum_{i=1}^{N_{states}} n_i \cdot I_i \cdot T_i \quad (3.11)$$

De igual manera el tiempo que se encuentra el dispositivo final en el estado de más bajo consumo o sleep se calcula como:

$$T_{sleep} = T_{notif} - T_{act} \quad (3.12)$$

siendo T_{act} la suma los estados relacionados con la transmisión que incluyen todos menos el T_{sleep} :

$$T_{act} = T_{rb} + T_{cgd} + T_{cfun1} + T_{cops} + T_{cgatt} + T_{ping} + T_{cfun0} \quad (3.13)$$

El tiempo de vida o lifetime en horas, $T_{lifetime_ACK}$, de una batería de capacidad $C_{battery}$ (expresada en mAh) se calculará como:

$$T_{lifetime} = \frac{C_{battery}}{I_{avg}} \quad (3.14)$$

Finalmente, el coste energético para un periodo completo, $EC_{delivery}$ se expresa siguiendo el mismo razonamiento que en los casos anteriores:

$$EC_{delivery} = \frac{I_{avg} \cdot V \cdot T_{notif}}{L_{delivery}} \quad (3.15)$$

siendo el numerador la energía consumida por el dispositivo durante T_{notif} . V indica el voltaje utilizado y $L_{delivery}$ indica el número de bits de información que se ha entregado de forma satisfactoria por cada trama enviada (payload).

CAPÍTULO 4. Resultados y discusión

En este capítulo se presentarán y discutirán los resultados obtenidos de las pruebas realizadas con los dispositivos correspondientes para Sigfox y NB-IoT, utilizando el modelo analítico presentado en el capítulo anterior.

Para cada tecnología se detallarán cuatro subapartados:

- Consumo de corriente del dispositivo final
- Tiempo de vida del dispositivo final
- Coste energético del envío de datos por cada bit útil entregado

Para el caso de Sigfox se tendrán en cuenta las transmisiones sin mensaje de bajada (sin ACK) y las transmisiones con mensaje de bajada (con ACK), para diferentes tamaños de payload. Para NB-IoT también se presentarán los resultados para varios payload.

4.1. Evaluación de Sigfox

En los siguientes subapartados se presentan los resultados obtenidos para Sigfox. Si bien el tiempo mínimo entre mensajes consecutivos sería de 10 minutos debido a la restricción de 140 mensajes uplink al día, también se muestran resultados para tiempos entre mensajes inferiores a 10 minutos puesto que bajo ciertas condiciones se podría operar sin restricciones de duty cycle, por ejemplo, en ciertas regiones del mundo donde no existen tales restricciones. Durante el desarrollo de las pruebas se ha utilizado un tamaño mínimo de payload de 1 byte y el tamaño máximo permitido de 12 bytes. En las pruebas de Sigfox, todos los mensajes fueron recibidos correctamente. Se desconoce la tasa de éxito de cada una de las 3 transmisiones individuales, así como la tasa de recepción individual de cada estación base.

4.1.1. Consumo de corriente

El consumo de corriente es uno de los parámetros de rendimiento más importantes de los dispositivos finales. De su magnitud depende la duración de la batería instalada y por lo tanto la vida útil del dispositivo.

Así pues, es un factor a tener en cuenta a la hora de desplegar una red ya que influye notablemente en la toma de decisiones a la hora, como en la elección de las baterías para conseguir un tiempo de vida determinado.

El consumo de corriente medio se ha calculado para los dos casos de transmisión como se indica en las ecuaciones (3.1) y (3.6). Se han realizado

diferentes pruebas para ver el consumo y duración de cada estado para asegurar los resultados, observando diferencias despreciables entre todas ellas.

En la siguiente gráfica se pueden observar los diferentes consumos de corriente asociados a cada modo de transmisión y tamaño de payload en función del tiempo entre notificaciones.

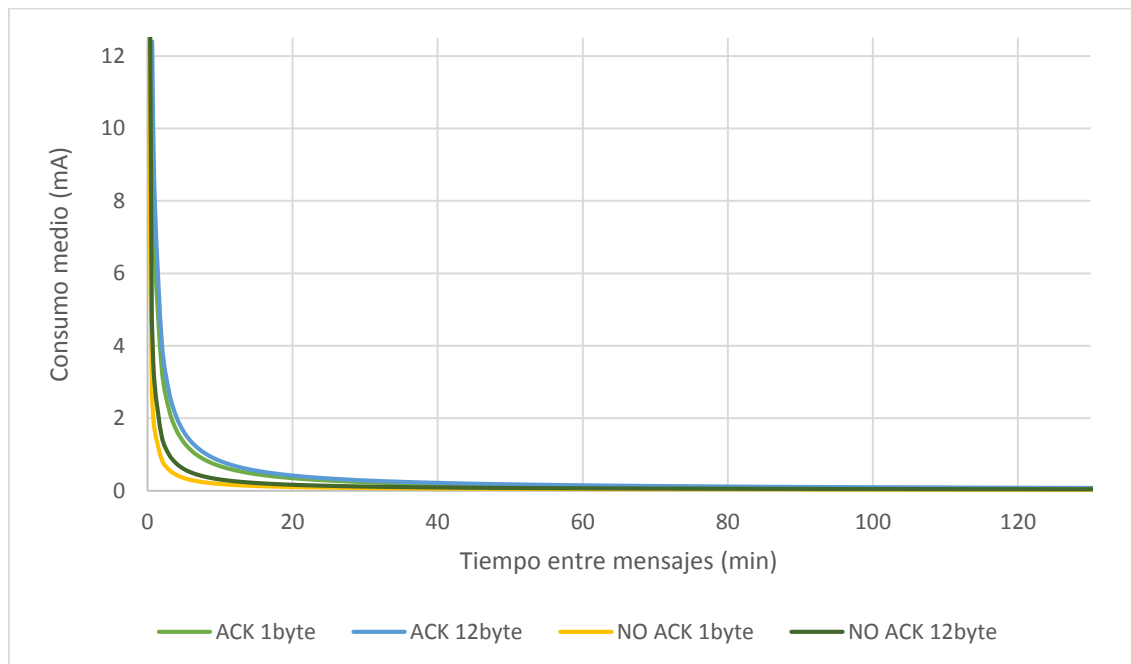


Fig. 4.1 Consumo de corriente medio para transmisiones Sigfox.

En la figura anterior se representa el consumo medio para los cuatro casos analizados: transmisiones con y sin ACK con tamaños 1 byte y 12 bytes.

Como se puede observar, la corriente media tiende asintóticamente hacia el consumo que se ha obtenido en el estado sleep ($\approx 16 \mu A$), que prácticamente se corresponde con el dato proporcionado por el fabricante en el datasheet [11] del ($12.8 \mu A$ a $25^{\circ}C$). Esto es debido a que, a medida que crece el tiempo entre mensajes, el dispositivo permanece más tiempo en este estado mientras se reduce el tiempo en los estados de mayor consumo.

Para las transmisiones con y sin ACK el comportamiento la curva de consumo es similar, alcanzando antes el mínimo consumo en el caso de transmisión sin ACK, debido a que únicamente se realiza el envío del mensaje 3 veces, sin procedimientos para la recepción de un mensaje de bajada y su posterior confirmación uplink. Por otro lado, el tamaño del payload también influye en el consumo de corriente, ya que el dispositivo se encontrará más o menos tiempo transmitiendo.

En la siguiente figura replicando la anterior, pero en escala logarítmica, se observa más detalladamente este hecho.

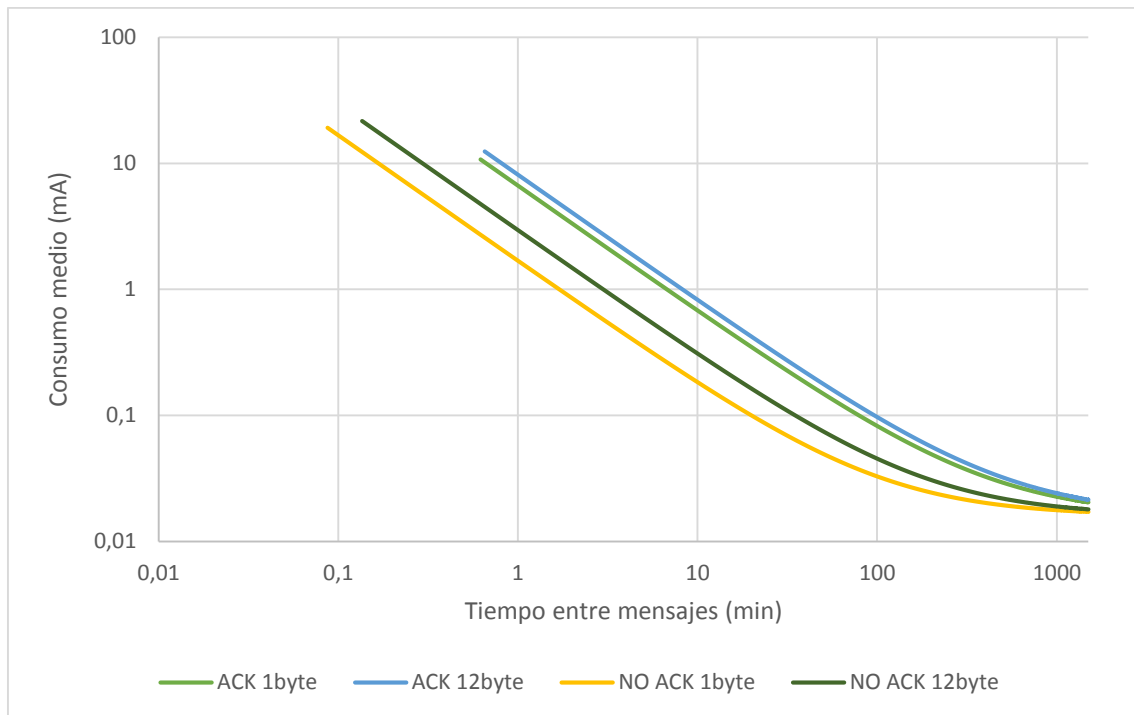


Fig. 4.2 Consumo de corriente medio para transmisiones Sigfox.

En la figura observamos con mayor detalle las diferencias cuando varía el payload de los paquetes transmitidos. Para el caso de no ACK las diferencias entre ambos payload son mayores que para el caso de ACK.

Esto es debido a que, en el primer caso, el tiempo en que está transmitiendo el dispositivo ocupa gran parte del periodo, mientras que en el otro caso encontramos el estado en que el dispositivo permanece en espera y escucha para recibir el mensaje de bajada.

Considerando el tiempo mínimo entre mensajes de 10 minutos para no sobrepasar los 140 disponibles al día, obtenemos un consumo mínimo de 18 mA (1 byte sin ACK) y un máximo de 82 mA (12 bytes con ACK).

Como se comentó en el capítulo 1, Sigfox puede transmitir a 100 bps o 600 bps, dependiendo de la zona geográfica de uso. En este caso únicamente se ha podido realizar las pruebas con la tasa transmisión de 100 bps. No obstante, podemos generar una versión de las figuras para 600 bps dividiendo el tiempo de transmisión por 6.

A continuación, se muestran las figuras generadas de consumo de corriente para 600 bps.

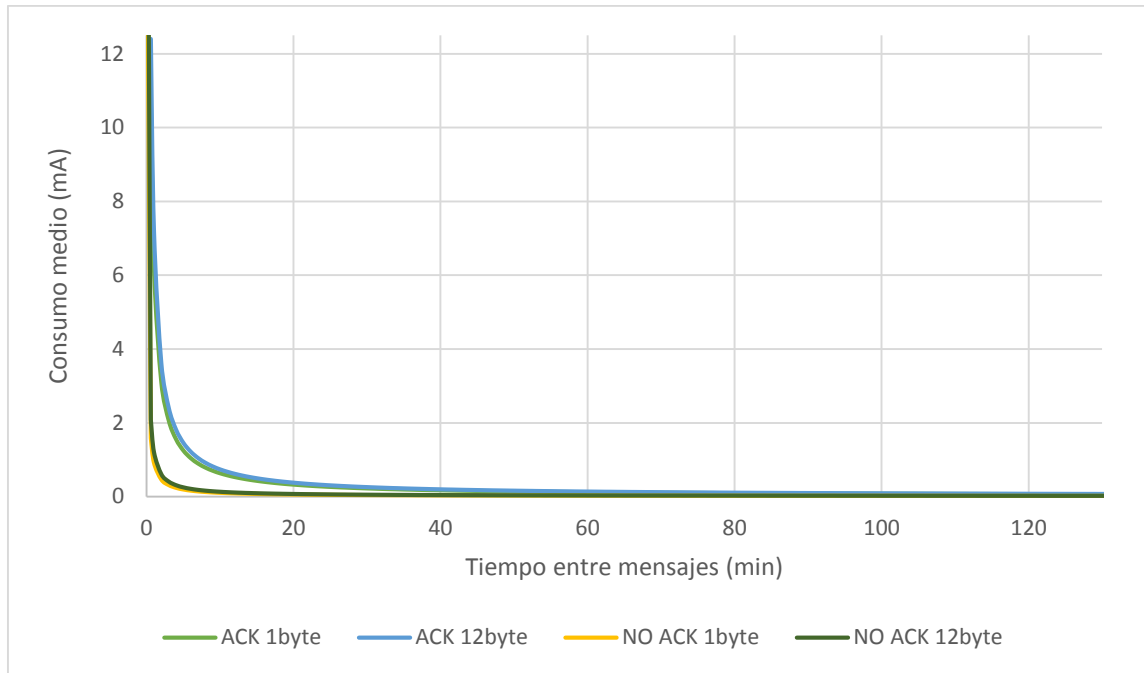


Fig. 4.3 Consumo de corriente medio para transmisiones Sigfox (600 bps).

Con esta velocidad de transmisión se observa como la curva se hace más acusada y se aproxima antes al consumo del estado sleep. También las diferencias de consumo entre los diferentes tamaños de payload disminuyen al reducirse el tiempo de transmisión. En escala logarítmica podemos observar mejor estas diferencias.

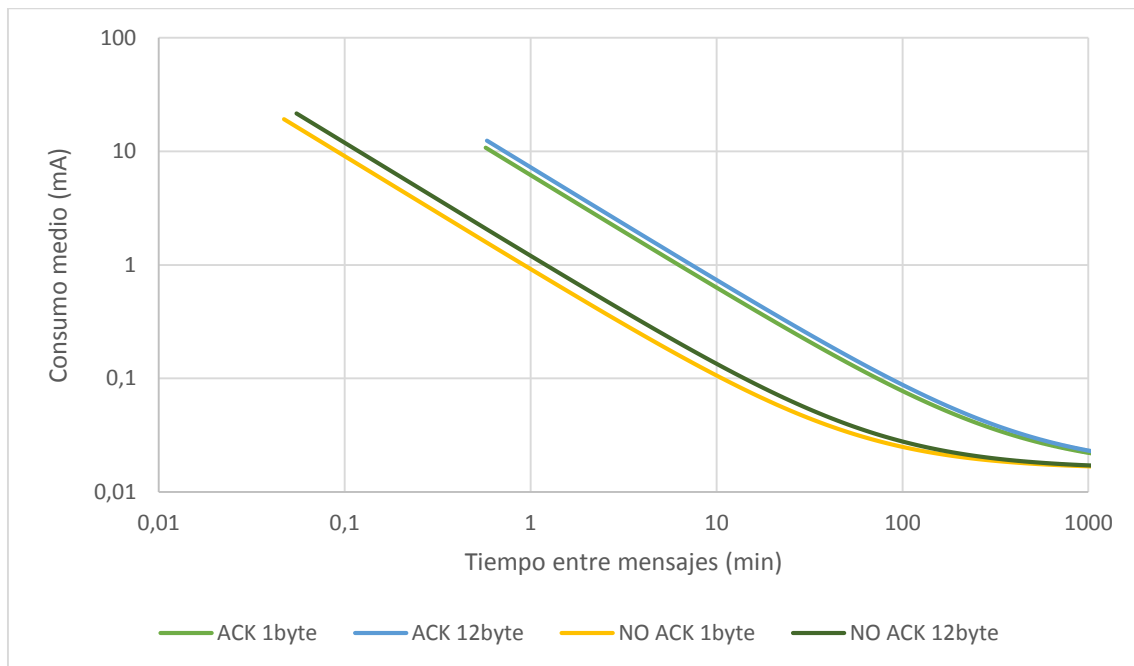


Fig. 4.4 Consumo de corriente medio para transmisiones Sigfox (600 bps).

4.1.2. Tiempo de vida

Aparte del consumo de corriente en las transmisiones, el tiempo de vida de los dispositivos depende en gran parte de la elección de la batería. Para calcular cuál es el tiempo de vida en ambos modos de transmisión, sin y con ACK, se han utilizado las ecuaciones (3.4) y (3.9), respectivamente. El tipo de batería utilizado en los cálculos ha sido de 3 V, con una capacidad de 2400 mA·h. Este tipo de batería se puede encontrar en el mercado fácilmente [18].

En la siguiente figura se muestra el tiempo de vida en años del dispositivo en función del tiempo entre transmisiones, para la batería mencionada.

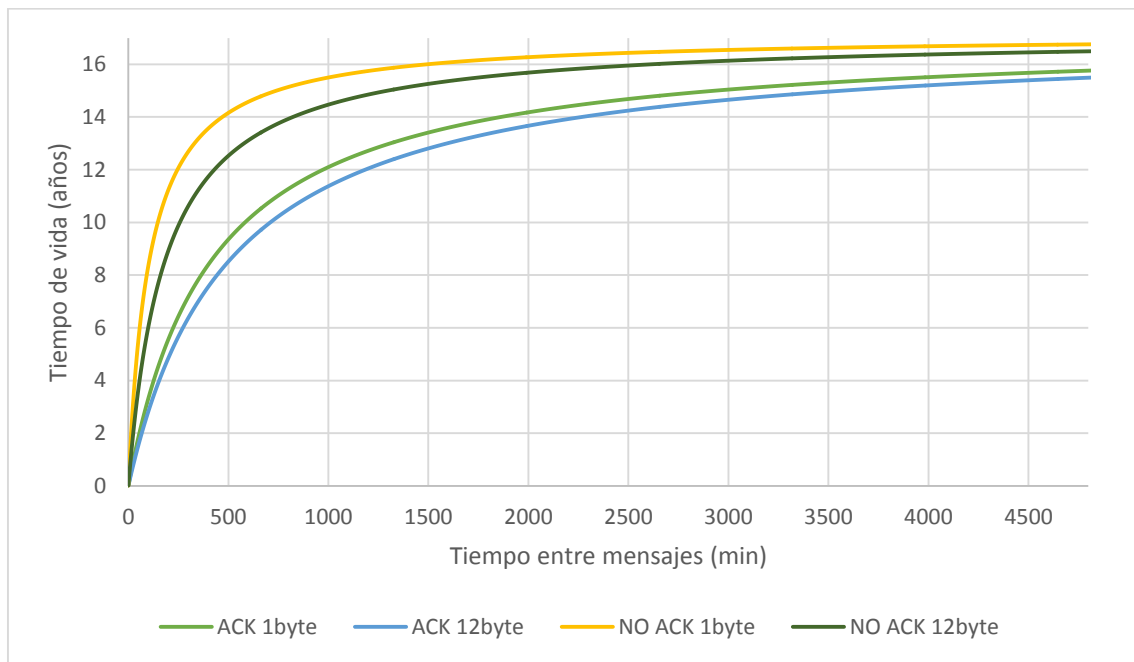


Fig. 4.5 Tiempo de vida del dispositivo en Sigfox.

Podemos observar claramente cómo el hecho de realizar transmisiones con ACK reduce claramente el tiempo de vida en comparación con el otro caso. Como es lógico, el tiempo de vida crece a medida que aumenta el tiempo entre mensajes, tendiendo asintóticamente sobre los 16 años.

Si se enviase un mensaje cada 24 horas, el tiempo de vida del dispositivo se encontraría por encima de los 12 años en el peor de los casos (ACK, 12 bytes) y casi 16 años para el mejor de los casos (no ACK, 1 byte). Para un tiempo entre mensajes de 10 minutos, obtendríamos un tiempo de vida mínimo de 4 meses (12 bytes con ACK) y un máximo de 18 meses (1 byte sin ACK).

La siguiente figura muestra los resultados obtenidos en escala logarítmica.

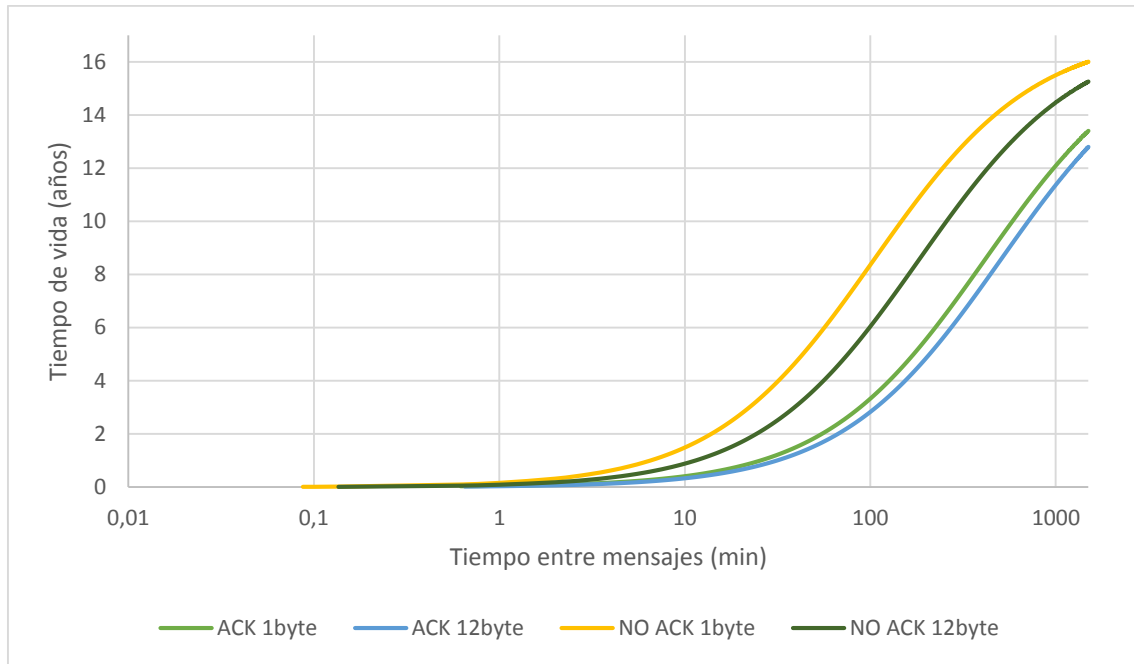


Fig. 4.6 Tiempo de vida para el dispositivo en Sigfox.

En la figura anterior podemos apreciar claramente cómo a partir de realizar transmisiones cada 10 minutos, el tiempo de vida comienza a crecer notablemente superando ya el año de vida para prácticamente todos los casos.

Para el estudio del tiempo de vida, también se han generado los resultados para la velocidad de transmisión de 600 bps.

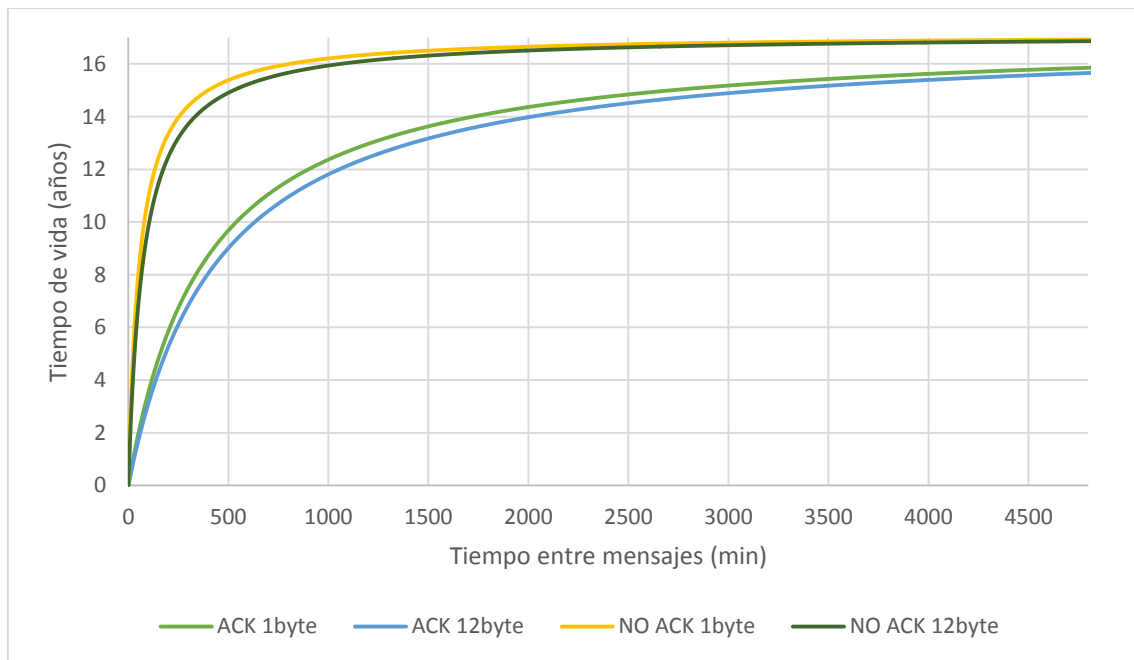


Fig. 4.7 Tiempo de vida para el dispositivo en Sigfox (600 bps).

Se puede observar cómo, utilizando esta velocidad de transmisión (600 bps), el tiempo de vida en general aumenta, así como las diferencias entre transmisiones con ACK y sin ACK, mientras que la influencia del tamaño de payload disminuye, por la razón de disminuir el tiempo en que el dispositivo se encuentra transmitiendo. En escala logarítmica podemos observar apreciar mejor estas diferencias.

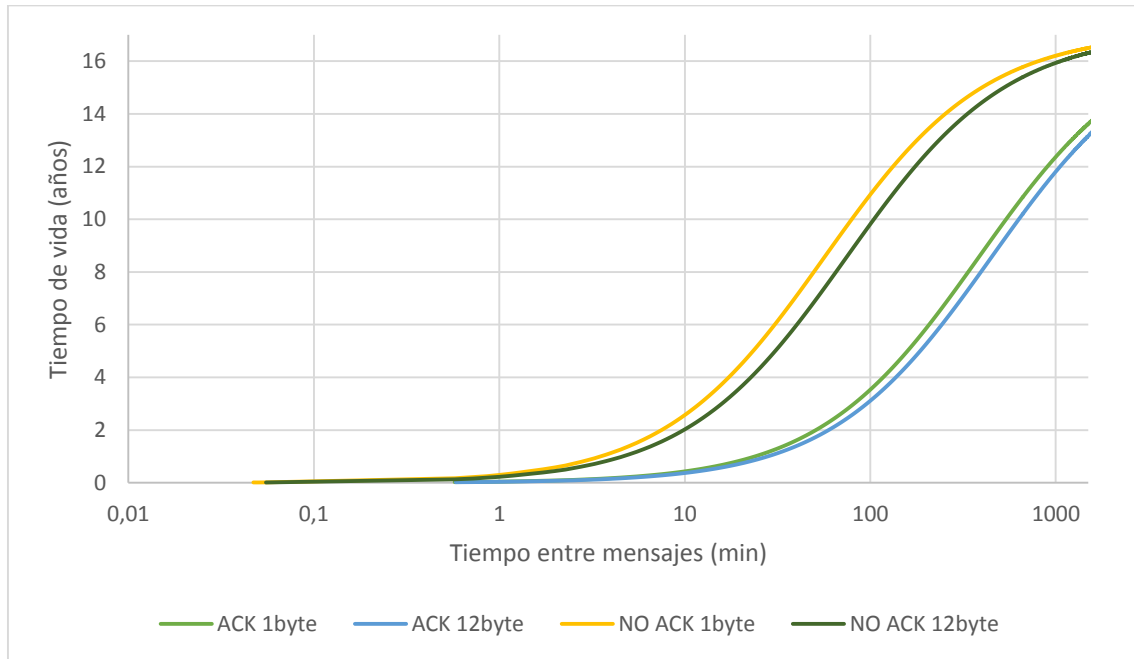


Fig. 4.8 Tiempo de vida para el dispositivo en Sigfox (600 bps).

4.1.3. Coste energético

En este apartado se evalúa otro de los parámetros de rendimiento más importantes como es el coste energético del envío de datos. Para calcular este parámetro en ambos modos de transmisión, sin y con ACK, se han utilizados las ecuaciones (3.5) y (3.10) respectivamente.

Se han asumido las características de la batería indicada en el apartado anterior. La siguiente figura muestra el coste energético para los cuatro casos analizados.

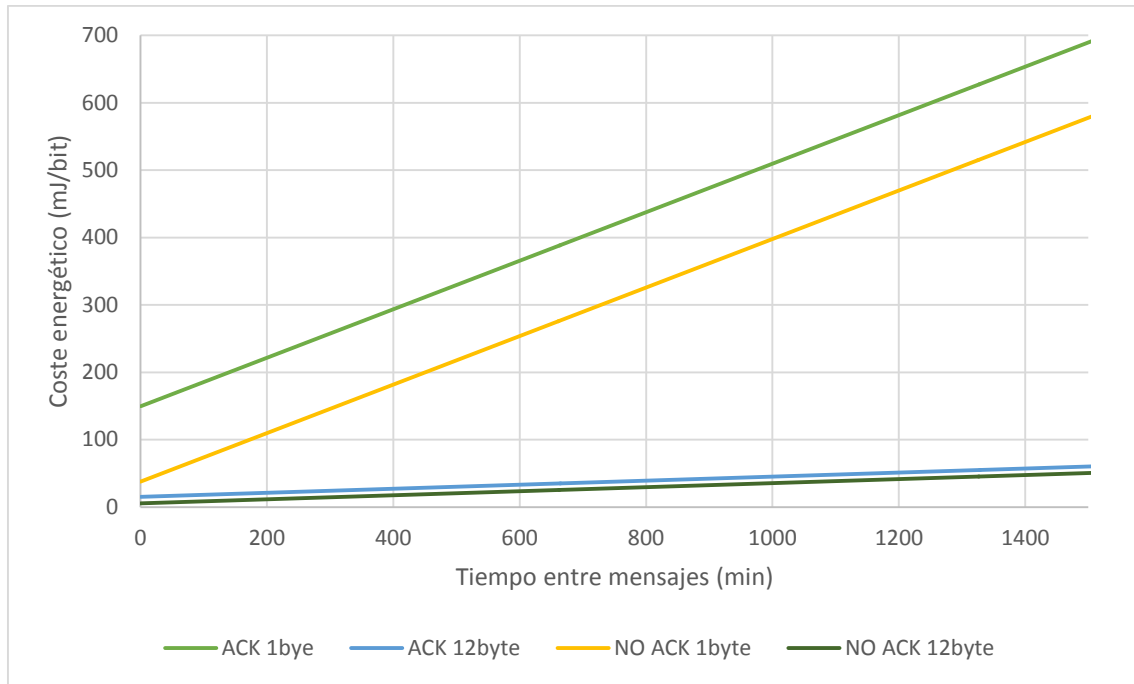


Fig. 4.9 Coste energético para transmisiones Sigfox.

En la anterior figura, vemos representada la energía requerida para entregar cada bit al destinatario. Este parámetro tiene una tendencia lineal creciente en función del tiempo entre transmisiones. Esto es debido a que, a medida que crece el tiempo entre mensajes, el consumo de corriente medio tiende al valor obtenido para el estado sleep. Por lo tanto, la energía media consumida durante el tiempo en que el dispositivo se encuentra en ese estado tendrá cada vez más influencia.

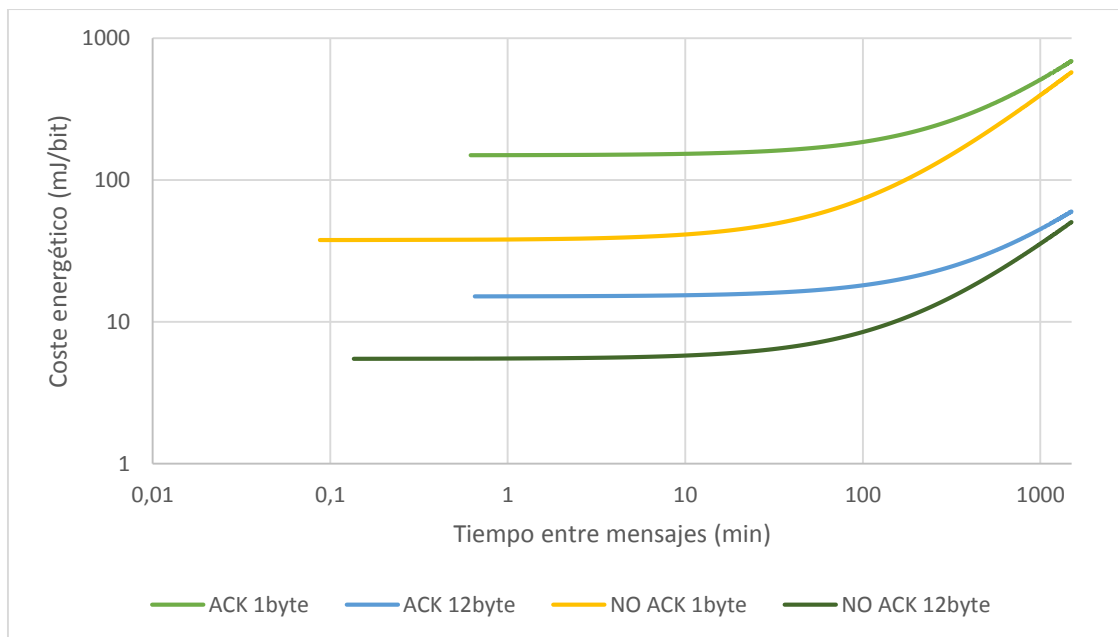


Fig. 4.10 Coste energético para transmisiones Sigfox.

La anterior figura muestra los resultados en escala logarítmica. Observamos que lo que tiene más peso sobre el coste energético es el tamaño de payload y no el modo de transmisión, que el coste energético es inversamente proporcional al payload enviado. Para un tiempo entre mensajes de 10 minutos, obtenemos un coste energético mínimo de 5,8 mJ/bit (12 bytes sin ACK) y un máximo de 153 mJ/bit (1 byte con ACK). A continuación, se presentan los resultados para la velocidad de 600 bps.

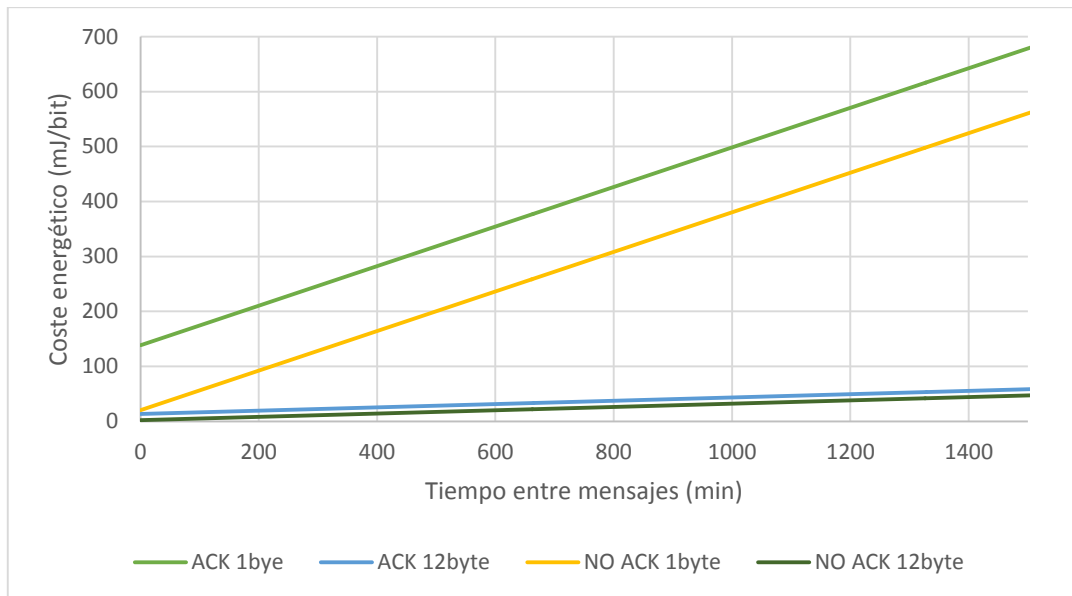


Fig. 4.11 Coste energético para transmisiones Sigfox (600 bps).

Como vemos en la figura anterior, utilizar una tasa de transmisión más alta, se traduce en reducir ligeramente el coste energético ya que el tiempo de transmisión global se reduce.

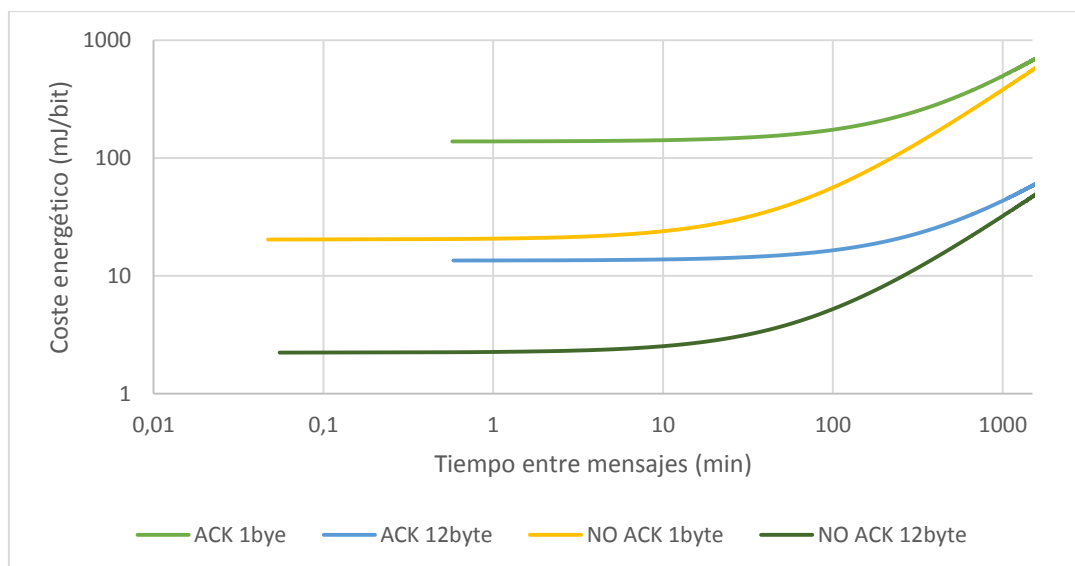


Fig. 4.12 Coste energético para transmisiones Sigfox (600 bps).

4.2. Evaluación de NB-IoT

En los siguientes subapartados se presentan los resultados obtenidos para NB-IoT. Durante el desarrollo de las pruebas no ha sido posible medir el consumo mínimo teórico de $5 \mu\text{A}$ anunciado en el datasheet del dispositivo [14], cuando éste se encontraba en modo PSM (ver 1.2.4).

Así pues, en la evaluación se ha considerado el valor del datasheet para el estado sleep, el período de configuración más envío de mensaje y diferentes tamaños de payload.

4.2.1. Consumo de corriente

El consumo de corriente medio se ha calculado utilizando la ecuación (3.11). Se han realizado diferentes pruebas para ver el consumo y duración de cada estado, y de esta manera poder ajustar el script para realizar todo el proceso satisfactoriamente en el menor tiempo posible.

En la siguiente gráfica se pueden observar los diferentes consumos de corriente asociados a cada tamaño de payload en función del tiempo entre notificaciones.

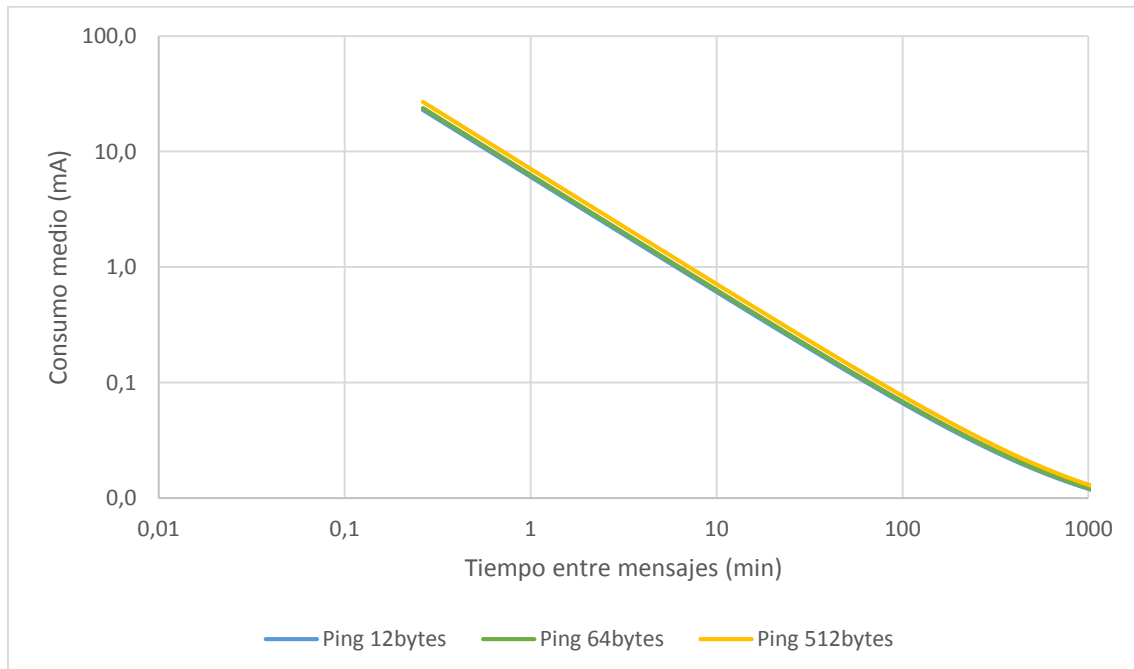


Fig. 4.13 Consumo de corriente medio para transmisiones NB-IoT.

Observamos en este caso cómo las diferencias de payload prácticamente no influyen en el resultado final, ya que lo que tiene más peso en términos de consumo es la configuración previa al envío del mensaje.

4.2.2. Tiempo de vida

Para calcular el tiempo de vida, se ha utilizado la ecuación (3.14). El tipo de batería utilizado en los cálculos ha sido en este caso de 5 V, con una capacidad de 2400 mA·h.

En la siguiente figura se muestra el tiempo de vida en años del dispositivo en función del tiempo entre transmisiones.

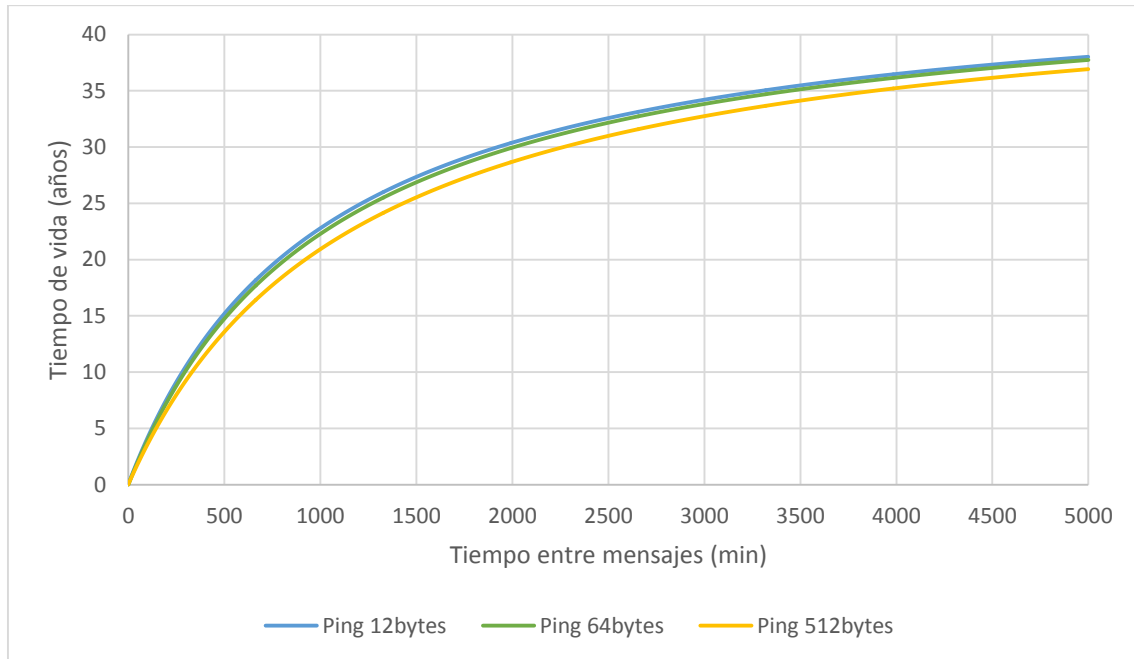


Fig. 4.14 Tiempo de vida para el dispositivo en NB-IoT.

En la figura anterior sí podemos observar una mayor diferenciación respecto al tiempo de vida en función del tamaño de payload de paquete. Vemos como en el caso de 512 bytes el tiempo de vida es ligeramente inferior al resto. Con un alto tiempo entre mensajes, la curva tiende a alcanzar el valor de ~40 años en tiempo de vida, una cifra poco realista por otro lado, ya que el dispositivo no utilizará una batería ideal.

4.2.3. Coste energético

Para terminar este capítulo, evaluamos en este apartado otro de los parámetros de rendimiento más importantes, como es el coste energético del envío de datos. Para calcular este parámetro se ha utilizado la ecuación (3.15). Para el cálculo, se han considerado las características de la batería indicada en el apartado anterior.

La siguiente figura muestra el coste energético para los tres casos analizados, en función del tiempo entre mensajes consecutivos.

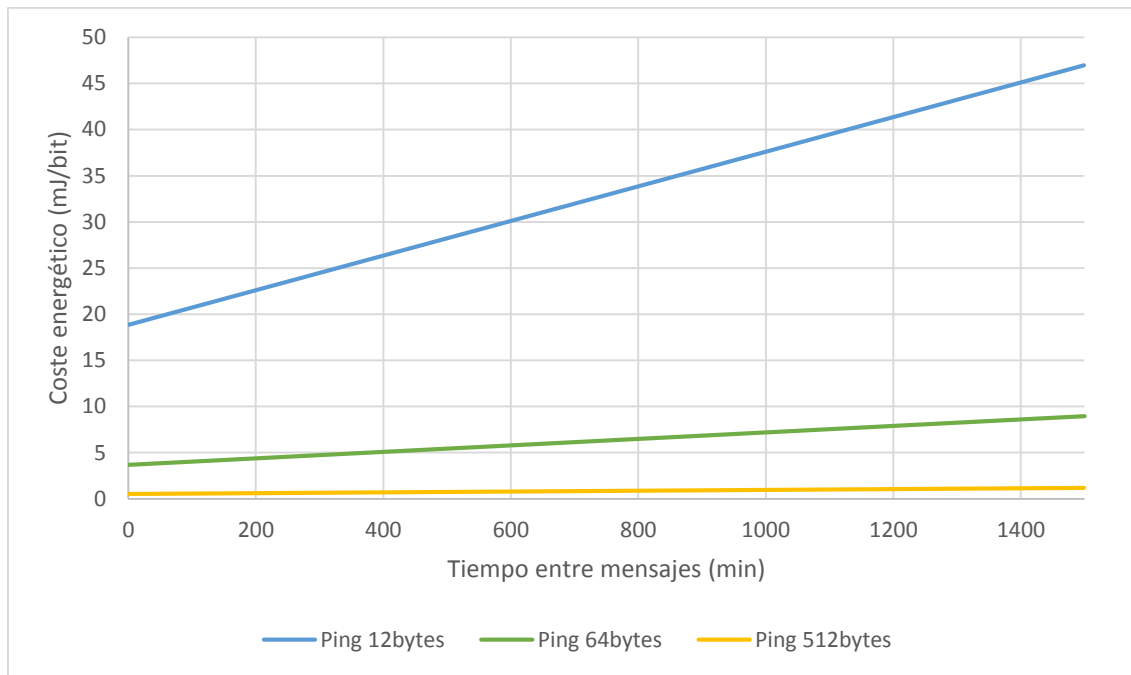


Fig. 4.15 Coste energético para transmisiones NB-IoT.

En la anterior figura vemos representada la energía requerida para entregar cada bit al destinatario. Observamos también una diferencia significativa en el caso de envíos de ping 12 bytes con respecto al resto de tamaños de payload ya que el coste energético es inversamente proporcional al payload enviado y por lo tanto, a medida que lo aumentamos aprovechamos más la transmisión y el coste energético decrece.

Vemos como la curva tiene una tendencia lineal creciente en función del tiempo entre transmisiones. De la misma manera que en Sigfox, esto es debido a que a medida que crece el tiempo entre mensajes el consumo de corriente medio tiende al valor obtenido para el estado sleep. Por lo tanto, la energía media perdida durante el tiempo en que el dispositivo se encuentra en ese estado tendrá cada vez más influencia.

CAPÍTULO 5. Conclusiones

En este último capítulo se presentarán las conclusiones y valoraciones obtenidas a partir de la realización de este trabajo. También se incluye un apartado de líneas futuras que se podrían tener en cuenta de cara a próximos proyectos, para seguir profundizando en el estudio de las tecnologías consideradas en este trabajo.

5.1. Conclusiones

El objetivo principal de este trabajo era realizar un estudio sobre las prestaciones en términos de energía de dos tecnologías LPWAN. En este caso, las tecnologías estudiadas fueron Sigfox y NB-IoT. Mediante medidas realizadas sobre un dispositivo final, se generaron modelos analíticos para evaluar las prestaciones de ambas tecnologías.

Para la realización de este trabajo se ha utilizado hardware específico para cada tecnología, así como software propietario y herramientas de laboratorio, con el fin de obtener una caracterización realista de los parámetros analizados. Este entorno de trabajo permitió realizar las pruebas necesarias para poder generar los modelos de consumo para cada tecnología.

En el caso de Sigfox, se han considerado diferentes tasas de transmisión durante el envío de datos con y sin ACK, así como diferentes tamaños de payload. Una cuestión clave era obtener un consumo en el estado sleep similar al estipulado en el datasheet para así poder obtener los resultados más realistas posibles sobre lo que se espera de una tecnología LPWAN. Sobre este aspecto podemos afirmar que se ha cumplido con el propósito.

Hemos visto cómo las transmisiones con ACK comportan un mayor consumo con respecto a las transmisiones sin ACK al tener un periodo extra de escucha y un envío adicional de mensaje uplink. También hemos observado que el tamaño de payload tiene más influencia en el caso de transmisiones sin ACK al estar más tiempo transmitiendo durante un periodo en tal caso.

Por lo que respecta al tiempo de vida de un dispositivo final, se han obtenido unos resultados que se pueden considerar aptos para diferentes aplicaciones IoT. Asumiendo una capacidad de batería de 2400 mA·h para obtener los resultados, observamos cómo a medida que aumenta el tiempo entre mensajes el tiempo de vida crece rápidamente. Por ejemplo, con una frecuencia de envío de 12 minutos se ha obtenido una duración de batería de más de un año.

Donde más se ha visto la influencia del tamaño de payload ha sido en el coste energético del envío de datos. La diferencia observada entre enviar el máximo payload permitido y el hecho de enviar 1 byte llega hasta dos órdenes de magnitud.

Para NB-IoT se consideraron diferentes tamaños de payload para evaluar su impacto. Al contrario que con Sigfox, en esta ocasión no se logró medir el consumo esperado en el estado sleep o ahorro de energía, tal y como se especificaba en el datasheet. Este hecho hizo que, para la evaluación, se asumiera el dato teórico de consumo de corriente en estado sleep proporcionado. Como consideración a la hora de establecer la duración de un periodo de actividad, se tuvo en cuenta la configuración del dispositivo en la red y el posterior envío de mensaje.

Hemos observado cómo el impacto del tamaño de payload no ha sido excesivamente relevante tanto en el consumo como en el tiempo de vida del dispositivo. Esto hecho es debido a que el consumo de corriente en los estados de configuración, previos al envío del mensaje, tienen más peso. Para cada intento registrado, se observaron ligeras diferencias de consumo y tiempo en dichos estados.

Respecto al tiempo de vida, también se han obtenido resultados que se considerarían aptos para diversas aplicaciones IoT, obteniendo por ejemplo un año de vida con una frecuencia de envío de un paquete ping (con su respuesta) cada 23 minutos. Finalmente, se aprecia más el impacto del tamaño de payload en el coste energético del envío de datos. La diferencia entre usar el tamaño de 12 bytes y el máximo considerado de 512 bytes es notable.

5.2. Líneas futuras

Consideramos que en el desarrollo de este trabajo se ha cumplido con los objetivos planteados al principio del mismo, en líneas generales. No obstante, hay diversos aspectos a tener en cuenta y que se podrían trabajar en un futuro.

Por ejemplo, para NB-IoT, sería interesante emplear un dispositivo que sí permitiese medir su consumo cuando este se encuentre en un estado de sleep o low power. De esta manera, se podrían realizar pruebas basadas completamente en datos experimentales reales. A nivel general, la información encontrada respecto al funcionamiento de NB-IoT no ha sido muy extensa ni clara. En un futuro próximo, cuando esta tecnología alcance mayor madurez, una posibilidad sería desarrollar modelos de energía más detallados.

Respecto a Sigfox, únicamente se ha podido trabajar con la tasa de transmisión de 100 bps. Obtener resultados empíricamente con la tasa de 600 bps podría ser una alternativa, así como explorar otras características de la red como la precisión de la localización de los dispositivos proporcionada por Sigfox.

Respecto a las pruebas realizadas, estas se han llevado a cabo en un entorno de laboratorio. En futuros proyectos, colocar diferentes dispositivos en el exterior para ver el impacto del resto del entorno en las transmisiones, o en lugares con menor cobertura, también podría ser interesante. De esta manera se podría generar un modelo en función de los resultados obtenidos en tales escenarios.

Bibliografía

- [1] Jiménez, M., “Sigfox, una de las armas de Europa en la batalla global por el internet de las cosas”, septiembre 2018. Recuperado de https://cincodias.elpais.com/cincodias/2018/09/14/companias/1536952461_446846.html
- [2] Aprendiendo Arduino – Arduino y Sigfox, marzo 2018. Recuperado de <https://aprendiendoarduino.wordpress.com/2018/03/05/arduino-y-sigfox/>
- [3] Mekkia, K., Bajic, E., Chaxel, F., Meyer, F., “A comparative study of LPWAN technologies for large-scale IoT deployment”, *ICT Express*, 2017.
- [4] Sigfox, “Sigfox Technical Overview”, Julio 2017. Recuperado de <https://www.ismac-nc.net/wp/wp-content/uploads/2017/08/sigfoxtechnicaloverviewjuly2017-170802084218.pdf>
- [5] Sinha, R.S., Wei, Y., Hwang, S.H., “A survey on LPWA technology: LoRa and NB-IoT”, *ICT Express*, 3, 14–21 (2017).
- [6] Afzal, J., “NB-IoT Deployment – What it Takes”, Marzo 2017. Recuperado de <https://www.netmanias.com/en/?m=view&id=blog&no=11743&xtag=iot-nb-iot&xref=nb-iot-deployment-what-it-takes>
- [7] IETF RFC 8376, “Low-Power Wide Area Network (LPWAN) Overview”, mayo 2018.
- [8] Quectel, “Power Saving”, Cap. 3 en BC95 Hardware Design, Rev. 1.3, 13-30, junio 2017.
- [9] Mouser Electronics, Arduino MKR FOX 1200. Recuperado de <https://www.mouser.es/new/arduino/arduino-mkrfox1200/>
- [10] Aprendiendo Arduino – Arduino MKRFOX1200, marzo 2018. Recuperado de <https://www.aprendiendoarduino.com/2018/03/05/arduino-mkrfox1200/>
- [11] Microchip, SAM D21 – Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM. Recuperado de <http://ww1.microchip.com/downloads/en/DeviceDoc/SAMD21-Family-DataSheet-DS40001882D.pdf>
- [12] AG, Arduino, Sigfox library, marzo 2017. Recuperado de <https://www.arduino.cc/en/Reference/SigFox>
- [13] Sixfab, NB-IoT Evaluation Board. Recuperado de <https://sixfab.com/product/nb-iot-evaluation-board/>
- [14] Quectel, “Electrical, Reliability and Radio Characteristics”, Cap. 5 en BC95 Hardware Design, Rev. 1.3, 13-30, junio 2017.

[15] Quectel, BC95 AT Commands Manual, Rev. 1.9, 13-30, diciembre 2017.

[16] Keysight Technologies, N6705A DC Power Analyzer, Modular, 600 W, 4 Slots. Recuperado de <https://www.keysight.com/en/pd-1123271-pn-N6705A/dc-power-analyzer-modular-600-w-4-slots?cc=ES&lc=eng>

[17] Pila Litio CR17450BL 3V 2400mah. Recuperado de https://www.hdibattery.com/p11802858_pila-litio-cr17450bl-3v-2400mah.html