PROYECTO FINAL DE MÁSTER

Máster en Ingeniería Telemática

IMPACTO DE MECANISMOS DE SEGURIDAD EN EL FUNCIONAMIENTO DE SENSORES IEEE 802.15.4

Autor:
CAROLINA TRIPP BARBA

Director:
Jordi Casademont Serra

Barcelona, 2009

“Es justamente la posibilidad de realizar un sueño
lo que torna la vida interesante”

Paulo Coelho

AGRADECIMIENTOS

Primero que nada, quiero agradecer a Dios por darme salud y serenidad en todo este tiempo lejos de mi casa. Por ayudarme a tomar buenas decisiones y por nunca dejarme sola.

Aníbal, por tu apoyo día a día, por escuchar mis quejas, animarme cuando sentí que no podía más con la presión. Te agradezco cada cosa que has hecho por mí, cada palabra, gracias por creer en mí cuándo ni yo misma lo hacía. Y sobre todo por acortar las distancias. Gracias por hacerme feliz.

A mi familia, a mi Papá y a mi Mamá que han soportado conmigo esta distancia. Papá sabes que sin ti no estaría aquí, gracias por apoyarme con mis locuras, no me alcanzarían nunca las palabras para decirte lo muchísimo que te admiro y te quiero. Mamá porque siempre estabas ahí tratando de hacerme sentir cerca y nunca dejarme sola, gracias por siempre tener tiempo para mí. Adrián, gracias por preocuparte por mí todos los días, por estar siempre al pendiente, eres el mejor hermano del mundo. Mis tías, primos y en general toda mi familia que siempre estuvieron ahí cuando lo necesitaba. A mis abuelos, papá Miguel aunque ya no estás conmigo, gracias por el apoyo el día que me vine, siempre creyó que yo podría lograr lo que me propusiera y aquí está la culminación de ese apoyo. Mamá Toya, gracias por todo el amor. Los quiero a todos y les agradezco toda su ayuda. Siempre me hacen falta.

A mi asesor, el Dr. Jordi Casademont Serra, toda la ayuda, supervisión y ánimos durante todo este año, gracias por su paciencia, por siempre tener las palabras justas, porque siempre te recibe con una sonrisa que te anima a seguir adelante.

A la Universidad Autónoma de Sinaloa y la Facultad de Informática Mazatlán, principalmente al rector y vicerrector de la zona sur, MC. Héctor Melesio Cuén Ojeda y MC. Rafael Mendoza Zataráin, que me dieron todo el apoyo necesario para iniciar mis estudios aquí en Barcelona bajo el programa Doctores Jóvenes en Áreas Estratégicas.

A los chicos de laboratorio 003, por su ayuda cuando no sabía qué hacer, por sus explicaciones en todo lo que necesité.

A mis compañeros, Karen, Paola, Felipe, Javi, Ernesto y Anabell. Por que estuvieron tanto en las horas de estudio y preocupación, como en las horas de festejo. Sobre todo eso, gracias por todos los festejos que nos hacían olvidar un poco las preocupaciones diarias que nos han llevado hasta esto.

Iván, aun estando lejos nunca dejé de molestarte, gracias por todos los favores.

Sin todos ustedes esto no estaría completo. Esto tal vez no sería lo que es.

Carolina

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

El fenómeno de las comunicaciones inalámbricas cada día está creciendo a gran escala, es una de las áreas de las telecomunicaciones que se está desarrollando de manera exponencial. El inicio de este fenómeno fue el hecho de desear reducir gastos de instalación pues se elimina todo lo referente a cableados, esto nos permite el intercambio de una gran cantidad de información con un mínimo de esfuerzo en este punto. Además de permitirse el hecho de agregar cada vez más dispositivos inalámbricos con una reducción considerable en costos.

Desde hace varios años ya, las comunicaciones inalámbricas pasaron a ser parte de la vida cotidiana, hasta el punto de que en la actualidad estamos completamente conectados, ya sea con nuestro móvil o cualquier otro dispositivo inalámbrico con el cual estamos en constante envío o recepción de información.

De los últimos avances en cuanto a tecnología inalámbrica se refiere fue la creación de dispositivos diminutos, baratos y de bajo consumo, que aun con estas características son capaces de enviar y procesar información vía radio. Estos dispositivos son los llamados sensores o motes; cuando una considerable cantidad de estos dispositivos trabajan juntos, es cuando se puede hablar de una Red Inalámbrica de Sensores [1].

Los avances actuales en las comunicaciones inalámbricas son las que han permitido que estos dispositivos puedan desarrollarse. Gracias al estándar IEEE 802.15.4 [2] es posible su conectividad, ya que define las características de la Capa Física (PHY) y la Capa de Acceso al Medio (MAC) para los dispositivos de área personal (LR-WPAN, Low-Rate Wireless Personal Area Networks). Cuyas principales características radican en la fácil instalación, confiable transmisión de información, corto rango de operación, bajo costo y una razonable duración de batería.

Sin embargo, este tipo de dispositivos es susceptible, al igual que las redes inalámbricas tradicionales, al ataque sobre la información transmitida. Es por ello que asegurar la información es una de las principales preocupaciones, ya que el canal de comunicación no requiere la participación física de un cable. Y debido a sus características (baja potencia, reducida capacidad de procesamiento y memoria) hacen muy difícil el uso de métodos criptográficos conocidos.

Por ello al considerar proveer seguridad, ya sea confidencialidad, integridad o autenticidad, se debe tener presente si esto tendrá un consumo extra de energía, el impacto en el uso en el ancho de banda, así como los retardos adicionales que añadirá a la comunicación.

Al usar algún algoritmo criptográfico se debe tener en cuenta algunas restricciones, como el tipo de transmisor usado, así como la frecuencia usada, ya que estas características varían entre dispositivos. Y dependiendo de ellos son las implementaciones que soportan.

En el presente trabajo, se presenta un estudio práctico del impacto de los mecanismos de seguridad en el funcionamiento de dispositivos que trabajan bajo IEEE 802.15.4, en comparación con un estudio teórico anterior [3].

Se trata de comparar los resultados obtenidos de manera teórica, con los resultados obtenidos en pruebas reales. Estructurándose de las siguientes secciones.

Capítulo 1. Introducción y breve explicación de los objetivos del proyecto.

Capítulo 2. El estado del arte, así como información sobre todo lo relacionado y utilizado para las pruebas de este proyecto.

Capítulo 3. Breve definición del problema, y lo que se desea alcanzar con esta investigación y análisis.

Capítulo 4. Aquí se presenta la metodología de las pruebas, ecuaciones y demás para la obtención de los resultados.

Capítulo 5. En este apartado se muestran los resultados del estudio práctico, lo cual es la finalidad del proyecto. Se presentaran tablas y gráficas con resultados.

Capítulo 6. Se presentan como conclusiones un breve análisis de los resultados.

Capítulo 7. Este capítulo finaliza con la presentación de algunas líneas de investigación tentativas en las cuales se pueden realizar futuras investigaciones.

2. BACKGROUND Y ESTADO DEL ARTE

Las redes inalámbricas de sensores consisten en gran cantidad de pequeños dispositivos, capaces de recoger todo tipo de información de su entorno, como son: temperatura, humedad, luz, movimiento, etc., a través de los sensores que llevan incorporados. Su reducido tamaño y la capacidad de transmitir sin cables, permiten un despliegue rápido y flexible de centenares de dispositivos.

Los últimos avances tecnológicos han hecho realidad el desarrollo de unos dispositivos diminutos, baratos y de bajo consumo, que además, son capaces tanto de procesar información localmente como de comunicarse de forma inalámbrica. La disponibilidad de microsensores y comunicaciones inalámbricas permite desarrollar redes de sensores para un amplio rango de aplicaciones.

Tomando como base lo anterior, una red de sensores se puede describir como un grupo de dispositivos o motes que se coordinan para llevar a cabo una aplicación especifica. Al contrario que las redes tradicionales, las redes de sensores llevarán con más precisión sus tareas dependiendo de lo denso que sea el despliegue y lo coordinadas que estén. Cada nodo de la red consta de un dispositivo con un microcontrolador, sensores y transmisor/receptor y forma una red con muchos otros nodos. Por sí mismo, un sensor es capaz de procesar una limitada cantidad de datos; pero cuando se coordina la información entre un importante número de nodos, éstos tienen la habilidad de medir o procesar información de mayor magnitud con gran detalle.

2.1. IEEE 802.15.4

El estándar 802.15.4 [2] define las especificaciones de la capa MAC (Control de Acceso al Medio) y de Capa Física (PHY) de los dispositivos inalámbricos de área personal (LR-WPANs). La última revisión corresponde al 2006. Dichas revisiones y actualizaciones son hechas por el grupo de trabajo 802.15.

Este estándar se enfoca principalmente en comunicaciones entre dispositivos de bajo costo y poca velocidad, para lograr un uso óptimo de la potencia disponible en la red. Éste fue concebido para comunicaciones con un radio de 10 metros y una velocidad de datos no superior a 250 Kbps

2.1.1. Capa física (PHY)

La capa física especifica diferentes frecuencias en las cuales se puede operar, según la revisión de 2006 estas pueden ser:

- ✧ 868 – 868.6 MHz (Europa)
- ✧ 902 – 928 MHz (América del Norte)
- ✧ 2400 – 2483.5 MHz (Alrededor del mundo)

El estándar define a nivel físico 27 canales de comunicación en los tres rangos de frecuencias distintos dentro de la banda ISM (Industrial, Científica y Médica), 16 Canales en la banda de 2.4GHz, 10 Canales en la banda de 915 MHz y 1 Canal en 868 MHz.

El canal 868 MHz, se usa exclusivamente en Europa a una tasa de datos de 20 Kbps, la banda de 915 MHz es usada en EEUU a una tasa de 40 Kbps y la banda de 2.4 GHz está disponible en todo el mundo y opera a la velocidad nominal de 250 Kbps. En la Tabla 1 podemos observar un resumen de las frecuencias y velocidades de transmisión de los datos mencionados.

Las modulaciones definidas en 802.15.4 son 2, BPSK y O-QPSK. La primera usada en el canal 868 MHz con una tasa de chip de 300 por segundo y en la banda de los 915 MHz con una tasa de chip de 600 cada segundo y la segunda en la banda de 2450 MHz. Opcionalmente podemos usar también las modulaciones ASK y O-QPSK para las bandas 868/915 MHz.

Entre las características que proporciona la capa física (PHY) están la activación y desactivación del transceptor radio, la detección de energía, el indicador de calidad del enlace (LQI, Link Quality Indicator), la selección de canal, la evaluación de canal libre (CCA, Clear Channel Assessment) y la transmisión y recepción de paquetes a través del medio físico.

Las redes IEEE 802.15.4 ofrecen direcciones de 16 o 64 bits, permiten topologías Mesh o en estrella y su ancho de banda depende de la frecuencia en la que se esté trabajando.

Tabla 1. Bandas de frecuencia y velocidades de transmisión de datos [2].

PHY (MHz)	Banda de frecuencia (MHz)	Parámetros de Spreading		Parámetros de Datos		
		Tasa de chip (Chips/s)	Modulación	Tasa de bit (Kbps)	Tasa de símbolo (ksímbolos/s)	Símbolos
868/915	868–868.6	300	BPSK	20	20	Binario
	902 - 928	600	BPSK	40	40	Binario
2450	2400–2483.5	2500	O-QPSK	250	62.5	16-nario Ortogonal

2.1.2. Capa de Acceso al Medio (MAC)

De acuerdo con el estándar, las principales tareas de la capa de acceso al medio son:

- ✧ Generación de beacons en los dispositivos que trabajan como coordinadores.
- ✧ Soportar asociación y disociación de redes de acceso personal, PAN (Personal Area Networks).
- ✧ Soportar seguridad en los dispositivos.
- ✧ Emplear mecanismo de CSMA/CA para acceso al medio.
- ✧ Manipulación y mantenimiento de mecanismos GTS.
- ✧ Proveer enlaces confiables entre entidades.

Una red IEEE 802.15.4 puede operar en modo “beacon” o en modo “sin beacon”, el cual se utilizó para realizar las pruebas en el apartado 5. Este ultimo modo, usa el protocolo CSMA/CA de la manera habitual. Cuando un dispositivo quiere transmitir datos el dispositivo espera un cierto número aleatorio de periodos de backoff. Luego comprueba si el medio está inactivo, en tal caso, se transmiten los datos; en caso contrario, se repite el proceso de backoff.

El modo “beacon” usa una estructura de supertrama que empieza con beacons mandados por un dispositivo dedicado denominado coordinador, a intervalos predeterminados entre 15 ms y 251 s. El tiempo entre estos beacons se divide en periodo de activo e inactivo. Durante el periodo inactivo, el dispositivo entra en un modo de baja potencia durante el cual no se puede transmitir datos, de modo que la comunicación solo tiene lugar durante el periodo activo. Este periodo activo se divide en 16 slots iguales y consiste en dos grupos: el Periodo de Acceso por Contienda (CAP) y un Periodo Opcional sin Contienda (CFP), que proporciona calidad de servicio a los datos. El acceso al canal durante el CAP es mediante CSMA/CA.

La transmisión de una trama es seguida por un tiempo entre tramas (IFS) para permitir a la capa de control de acceso al medio un espacio finito para procesar los datos recibidos por la capa física. Antes de empezar el período de backoff, el dispositivo espera un IFS. Tras la transmisión de una trama larga (> 18 bytes) sigue un IFS largo (LIFS), mientras que tras la transmisión de una trama corta (< 18 bytes) sigue un IFS corto (SIFS). Si no se usan Acknowledgments (ACK’s), el IFS sigue inmediatamente después de la trama. Esto se puede ver en la Figura 1.

La capa de Control de Acceso al Medio define una dirección para el dispositivo de 64 bits, o bien una dirección corta de 16 bits, en el presente proyecto se ha hecho uso de las direcciones largas. Estas son asignadas durante el periodo de asociación

del dispositivo a la red. El tamaño del campo de dirección (que puede incluir tanto origen como destino) puede variar de 0 a 20 bytes, dependiendo si se toma direcciones de 16 ó 64 bits, mientras que las tramas de ACK no llevan información de dirección. Adicionalmente el campo de dirección puede contener un identificador de PAN de 16 bits, tanto para el emisor como del receptor. Estos identificadores solo pueden ser omitidos cuando no se mandan direcciones.

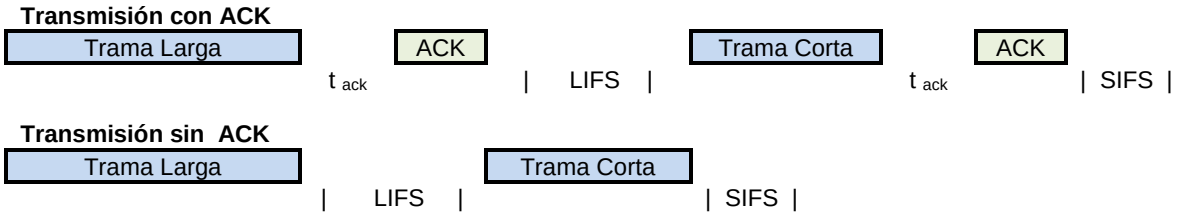

Figura 1. Tiempos IFS.

El campo de datos de la capa de Control de Acceso al Medio es variable, con la limitación de que el tamaño máximo de la trama (incluyendo cabecera) no puede exceder 127 bytes. Este formato de trama (especificado en el estándar IEEE 802.15.4) se puede ver a continuación en la Figura 2.

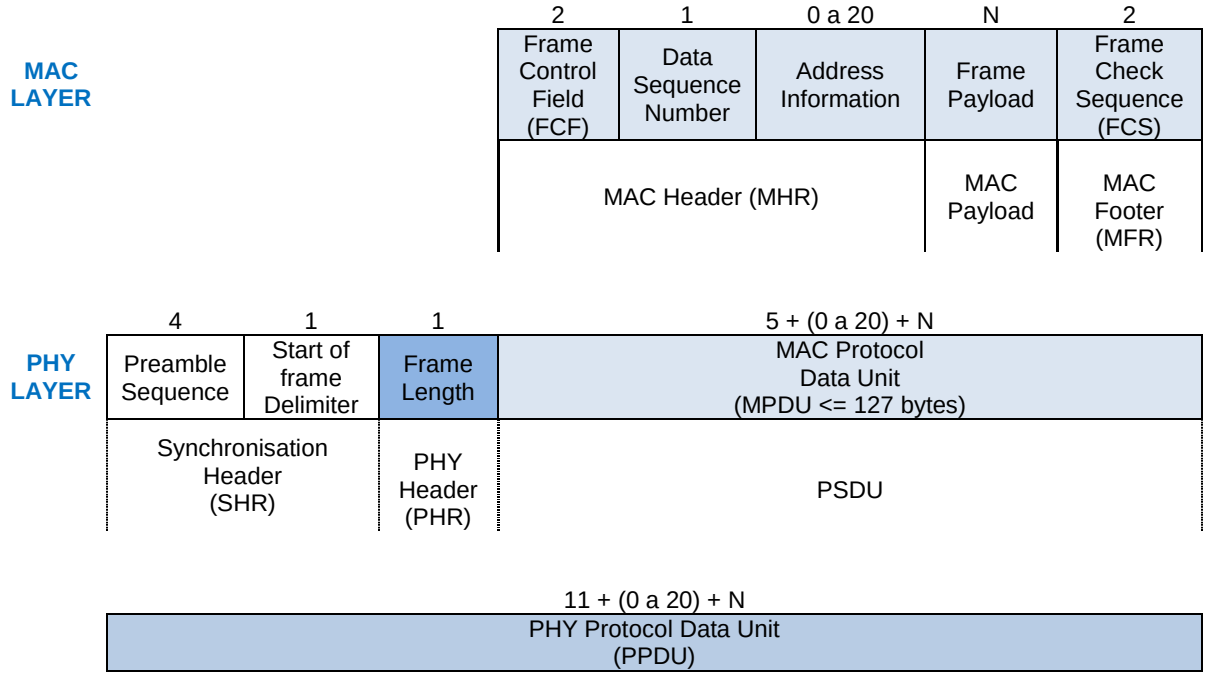

Figura 2. Formato de trama del estándar IEEE 802.15.4 (tamaños expresados en bytes).

La unidad de datos a nivel físico (PPDU) que se ilustra en la Figura 2 está compuesta por tres partes:

- ✧ SHR, permite a un dispositivo receptor sincronizarse para poder leer bien la información contenida en la PPDU, también indica el final de trama, ya que la trama puede tener una longitud variable.

- ✧ PHR indica la longitud de información ya que esta puede ser variable como hemos comentado anteriormente.

- ✧ PSDU es la información de la trama a nivel físico.

El campo de control o Frame Control Field (FCF) tiene un tamaño de 16 bits y contiene información como el tipo de trama, campos de dirección, Flags de control como el de seguridad y ACK, entre otros. La Figura 3 muestra su formato.

Bits: 0 - 2	3	4	5	6	7 – 9	10 - 11	12 - 13	14 - 15
Frame Type	Security Enabled	Frame Pending	ACK request	Intra PAN	Reserved	Destination Addressing Mode	Reserved	Source Addressing Mode

Figura 3. Formato del campo de control.

El estándar IEEE 802.15.4 se ha desarrollado para entornos hostiles y medios compartidos por lo que se han definido una serie de mecanismos para que sea robusto en estos entornos:

- ✧ CSMA/CA: sistema anteriormente comentado basado en la detección de portadora evitando colisiones. Su esquema de funcionamiento lo hace excelente compartiendo el medio.

- ✧ Tramas con confirmación (ACK): cuando enviamos tramas, se nos devuelve una trama ACK confirmando que la trama de datos o cualquier otra ha sido recibido correctamente.

- ✧ Verificación de los datos (CRC): mediante un polinomio generador de grado 16 se obtiene la redundancia y se puede comparar el CRC enviado con el calculado en destino para verificar los datos.

- ✧ Restricciones de consumo: IEEE 802.15.4 está pensado para aplicaciones que utilicen una batería o una unidad de energía agotable. Estas aplicaciones transmitirán información de forma muy esporádica por lo que la cantidad de energía que consume un nodo cuando escucha el canal es ultra baja.

- ✧ Seguridad: implementa seguridad de clave simétrica mediante el estándar de encriptación AES [4].

2.2. TelosB

Una plataforma hardware utilizada como nodo de red IEEE 802.15.4, es un dispositivo desarrollado y distribuido por la Universidad de California, Berkeley y Crossbow Technology Inc., denominada Telos revisión B.

Crossbow's TelosB [5] es un módulo inalámbrico de muy bajo consumo de potencia. Los módulos poseen estándares conocidos como USB y IEEE 802.15.4 para interoperar de forma transparente con otros dispositivos. Gracias al uso de estos estándares y periféricos integrados que miden la humedad, intensidad de luz y temperatura, Telos permite una gran variedad de aplicaciones para redes Mesh. La revisión B incluye mejoras en las funcionalidades y posibilidades de expansión. Gracias a la asociación estratégica con TinyOS, Telos es ideal para probar protocolos inalámbricos emergentes y para el movimiento de software de código abierto.

Figura 4. Mote TelosB.

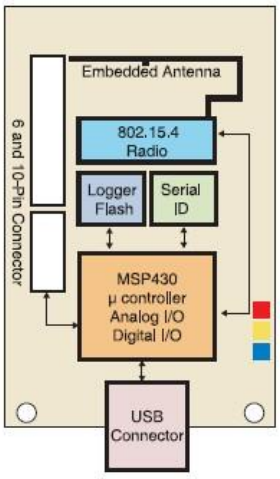

Figura 5. Diagrama de bloque.

Entre sus características principales se pueden encontrar las siguientes:

- ✧ Transceptor inalámbrico Chipcon IEEE 802.15.4.
- ✧ Compatibilidad global con las bandas ISM (2.4 hasta 2.4835 GHz).
- ✧ Velocidad 250 Kbps.
- ✧ Interoperabilidad con otros dispositivos IEEE 802.15.4.
- ✧ Microcontrolador MSP430 de Texas Instruments. 8 Mhz (10k RAM, 48k Flash).
- ✧ Convertidores AD y DA integrados.
- ✧ Antena integrada con rango de hasta 125 m.
- ✧ Consumo de corriente muy bajo.
- ✧ Programación y recolección de datos vía USB.
- ✧ Cambio rápido desde el modo descanso (< 6us).
- ✧ Sensores varios integrados. Luz, temperatura y humedad.
- ✧ Soporte de TinyOS.

Los nodos Telos pueden ser alimentados con 2 baterías AA. El módulo fue diseñado para encajar en el factor de forma de las 2 baterías. Estas pueden ser usadas en el rango operativo de 2.1 a 3.6 Voltios DC, sin embargo el voltaje debe ser al menos de 2.7 V cuando se programa el microcontrolador. Si el módulo Telos es conectado al puerto USB para ser programado, recibirá toda la potencia del ordenador. El voltaje de operación cuando se encuentra conectado a un ordenador es de 3 V, y durante todo el tiempo que permanezca conectado no usará potencia alguna de las baterías, si es que se encuentran conectadas. Si el módulo recibe una cantidad superior a 3.6 V puede desencadenar daños irreversibles sobre el microcontrolador, radio u otros componentes.

2.3. TinyOS

TinyOS [6] fue el primero sistema operativo basado en eventos diseñado para redes de sensores inalámbricas. El diseño de TinyOS está basado en responder a las características y necesidades de las redes de sensores, tales como el reducido tamaño de memoria, bajo consumo de energía, operaciones de concurrencia intensiva (simultaneidad en la ejecución de múltiples tareas interactivas). Por ello se encuentra optimizado en términos de uso de memoria y eficiencia de energía.

Fue desarrollado por la Universidad de California, Berkeley. Es código abierto, por lo cual sigue en desarrollo, lo cual provoca grandes beneficios para los clientes pues pueden adaptarlos a sus necesidades.

Su desarrollo se realizó como una necesidad de equipar los nodos del proyecto SmartDust de la misma universidad con un sistema operativo adaptado a sus posibilidades de hardware. El resultado fue presentado en el congreso Architectural Support for Programming Languages and Operating Systems 2000 (ASPLOS), y actualmente es la opción más generalizada para equipar los nodos de una WSN.

El diseño del Kernel (núcleo) de TinyOS está basado en una estructura de dos niveles de planificación.

- ✧ Eventos: Pensados para realizar un proceso pequeño (por ejemplo cuando el contador del timer se interrumpe, o atender las interrupciones de un conversor análogo-digital). Además pueden interrumpir las tareas que se están ejecutando.

- ✧ Tareas: Las tareas están pensadas para hacer una cantidad mayor de procesamiento y no son críticas en tiempo. Las tareas se ejecutan en su totalidad, pero la solicitud de iniciar una tarea, y el término de ella son funciones separadas.

Aunque también se hace uso de comandos, que son las llamadas a otros componentes de capas inferiores.

Este diseño permite que los eventos (que son rápidamente ejecutables), puedan ser realizados inmediatamente, pudiendo interrumpir a las tareas (que tienen mayor carga computacional en comparación a los eventos). Para el presente proyecto se usó la versión 2.0.2 [7] de dicho sistema operativo. Y su funcionamiento general puede observarse en la Figura 6.

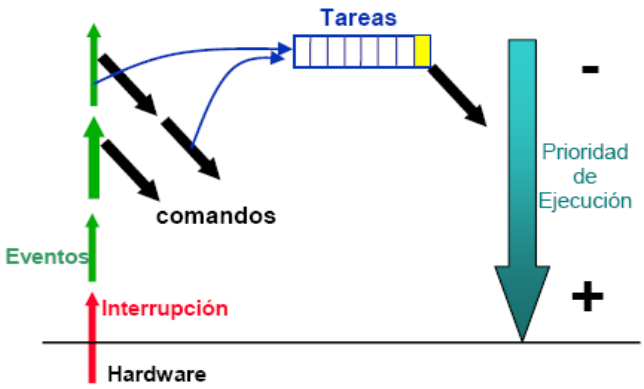

Figura 6. Esquema de funcionamiento de TinyOS.

2.4. NesC

La programación de sensores es relativamente compleja, entre otras dificultades está la limitada capacidad de cálculo y la cantidad de recursos. Y así como en los sistemas informáticos tradicionales se encuentran entornos de programación prácticos y eficientes para generar y depurar código, incluso en tiempo de ejecución, en estos microcontroladores todavía no hay herramientas comparables.

NesC [8] (Network Embedded Systems – C) es una extensión del lenguaje C para funcionar bajo sistemas integrados y concretamente como opción utilizada en la programación de los módulos que forman el sistema operativo funcionando bajo TinyOS. Es un lenguaje de programación basado en C, orientado a eventos. Utiliza un modelo de programación que integra el manejo de comunicaciones, la capacidad de reaccionar frente a sucesos (eventos) que puedan ocurrir durante la ejecución del programa.

Básicamente NesC ofrece la separación entre la construcción (módulo) y la composición (configuración). Los componentes, que son fragmentos de código, pueden ser de dos tipos: módulos y configuraciones, como se mencionó hace un momento. Los módulos proveen el código de la aplicación, implementando los eventos de una o más interfaces. Las interfaces son una agrupación de comandos y eventos. Mediante los eventos de una interface, se da respuesta a un suceso determinado, como por ejemplo realizar una determinada función cuando se recibe un mensaje, desde el componente en el que se usa la interface. Estas interfaces son los únicos puntos de acceso mediante sus comandos al resto de componentes que intervengan en la aplicación

En resumen, una Interfaz declara los servicios que se proveen y los servicios que se usarán, un Módulo provee el código de la aplicación implementando una o más interfaces y la Configuración declara la manera en que se unirán los distintos componentes, es decir, realiza el control de flujo.

Los tipos de datos implementados en NesC son el conjunto de datos disponibles en C, más un conjunto de tipos que aportan información útil para el conocimiento rápido del tamaño de dichos datos. Se puede encontrar uint8_t (entero sin signo de 8 bits), uint16_t (entero sin signo de 16 bits), result_t (booleano que puede tomar valores de SUCCES o FAIL) y bool (booleano que puede tomar valores de TRUE o FALSE). Además existen un conjunto de funciones específicas al lenguaje NesC, estas son: task, command y event

La función task, permite indicar una tarea que debe realizarse concurrentemente al funcionamiento de la aplicación, eso significa que cuando se inicia una tarea, esta se ejecuta hasta su finalización y después continuará el desarrollo del programa. Si ya existiera una tarea en ejecución esta seria puesta en una cola FIFO (First In – First Out), ejecutándose una a continuación de otra. Una tarea solo podrá ser interrumpida por una interrupción hardware, las interrupciones poseen la prioridad más elevada, y en su ejecución son pausadas las tareas en

ejecución hasta la finalización del código indicado en la interrupción. Las funciones command y event son específicas para comunicarse con otros componentes, así la ejecución de un command implica código disponible en otro módulo. El comando event es señalado por otro componente para indicar al presente componente que se ha producido una acción que requiere ejecución de su código.

2.5. CC2420

El SmartRF CC2420 [9] es un chip IEEE 802.15.4/ZigBee, operando en la banda de 2.4 GHz con velocidad de datos de 250 Kbps. Es actualmente uno de los chips más populares para trabajar en redes inalámbricas de sensores. Trabaja a bajos costes, con bajos voltajes y con aplicaciones inalámbricas de bajo consumo.

Entre sus aplicaciones se encuentran:

- ✧ Los sistemas IEEE802.15.4, que trabajan a 2.4GHz.
- ✧ Sistemas ZigBee.
- ✧ Automatización de hogares y edificios.
- ✧ Control industrial.
- ✧ Redes inalámbricas de sensores.
- ✧ Periféricos para PC.
- ✧ Aparatos electrónicos.

Figura 7. Chip CC2420 [9]**.**

Además incluye el soporte hardware de las siguientes características de la capa MAC del estándar IEEE 802.15.4:

- ✧ Generación automática de preámbulos.
- ✧ Detección e inserción de sincronización.
- ✧ Generación y comprobación de CRC sobre el payload.
- ✧ Evaluación de canal libre, CCA (Clear Channel Assesment).
- ✧ Detección de energía.
- ✧ Indicación de calidad de enlace.
- ✧ Seguridad.

CC2420, tiene el soporte de hardware para el formato de trama del estándar 802.15.4, que se muestra en la Figura 2. Además de permitir el uso de tramas ACK, cuyo formato es el de la especificación del estándar que se muestra en la Figura 8.

Bytes: 4	1	1	2	1	2
Preamble Sequence	Start of Frame Delimiter (SFD)	Frame Length	Frame Control Field (FCF)	Data Sequence Number	Frame Check Sequence (FCS)
Synchronization Header (SHR)		PHY Header (PHR)	MAC Header (MHR)		MAC Footer (MFR)

Figura 8. Formato de trama ACK [2]**.**

Como se mencionó en el listado anterior, una de las principales características del chip CC2420 es el de soportar operaciones de seguridad, como cifrado, autenticación e integridad. Es capaz de realizar dichas operaciones a nivel MAC, entre las cuales se incluyen CTR (cifrado), CBC-MAC (autenticación e integridad) y CCM (cifrado + autenticación e integridad). Cada unas de estas basadas en el cifrado AES (Advanced Encryption Standard) usando claves de 128 bits.

Dentro del espacio RAM de CC2420 se almacenan dos claves individuales, las cuales son identificadas como KEY0 y KEY1. La manera de establecer el uso de ellas puede hacerse por cada aplicación en particular, pues el estándar IEEE 802.15.4 no define como debe hacerse esto, se deja para las capas superiores del protocolo.

2.5.1. Seguridad In-line

CC2420 puede realizar operaciones de seguridad (cifrado, descifrado, autenticación e integridad) a nivel MAC dentro de las tramas TxFIFO y RxFIFO. Estas operaciones son llamadas operaciones de seguridad In-line [9].

Como otros soportes de hardware que tiene CC2420, las operaciones de seguridad In-Line afectan en el tamaño del campo de la cabecera Física (PHY). La longitud correcta de este campo debe usarse dependiendo la operación de seguridad a implementar.

Debe considerarse tanto la clave, *nonce* (que no aplican en caso de CBC-MAC) que son los datos extras usados para seguridad y los registros necesarios antes de iniciar a trabajar con ella. Los registros de seguridad son SECCTRL0 y SECCTRL1, los cuales deben activarse antes de iniciar con dichas implementaciones.

El formato de *nonce* puede verse en la Figura 9, el cual incluye los campos de contador de secuencia de claves que incrementa en cada trama enviada y el campo de contador de trama, los cuales como se verá más adelante afectan en el payload de las tramas.

Bytes: 1	8	4	1	2
Flags	Source Address	Frame Counter	Key Sequence Counter	Block Counter

Figura 9. Formato de Nonce de IEEE 802.15.4 [2]**.**

Los distintos modos de trabajar son:

- ✧ Sin seguridad
- ✧ CTR (cifrado / descifrado)
- ✧ CBC – MAC (autenticación e integridad)
- ✧ CCM (autenticación, integridad y cifrado / descifrado)

Sin Seguridad. Este modo constituye la función identidad y no proporción ningún tipo de seguridad.

CTR. Este modo proporciona confidencialidad usando el algoritmo de cifrado por bloques AES. El emisor rompe el texto en claro en bloques de 16 bytes y computa la operación $c_i = p_i \oplus E_k(x_i)$, siendo p_i un bloque de texto en claro, c_i el correspondiente texto cifrado, y x_i el valor de un contador empleado para el bloque *i*-ésimo.

Es decir, se usa un cifrado de bloque para producir un flujo pseudoaleatorio conocido como keystream. Este flujo se combina con el texto plano mediante XOR dando lugar al cifrado.

Los campos que se agregan en este mecanismo son un contador de trama (CT) de 4 bytes, que identifica la trama a transmitir y un campo de contador de clave

(CC) de 1 byte, que es incrementado en el caso de que el contador de trama llegue a su valor máximo. Usando este modo, el emisor incluye el contador de trama y el contador de clave, junto con los datos cifrados en el campo de datos de la trama a transmitir.

Cuando se usa este tipo se seguridad solo el payload es cifrado. Cuyo tamaño no varía, ya que es igual antes y después de ser cifrado.

Como desventajas hay que tener en cuenta que reutilizar un contador en la misma clave puede ser desastroso, pues se generará de nuevo el mismo keystream.

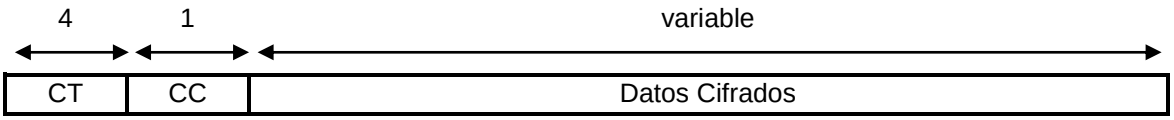

Figura 10. Campo de datos de una trama IEEE 802.15.4 con modo de seguridad CTR

CBC-MAC. Este tipo proporciona integridad y autenticación mediante el algoritmo CBC-MAC. El emisor puede generar un código MAC de 4, 8 ó 16 bytes. El código MAC puede ser calculado por entidades que compartan una misma clave simétrica. Este código protege tanto a los campos de datos como a las cabeceras de la trama a transmitir. El emisor añade el código MAC al texto en claro. El receptor verifica el código MAC calculando el código a partir de la trama recibida y comparándola con el valor incluido en dicha trama. La comprobación de este código permite verificar que los datos no fueron alterados, que el mensaje proviene del remitente correcto y que el mensaje sigue una secuencia correcta.

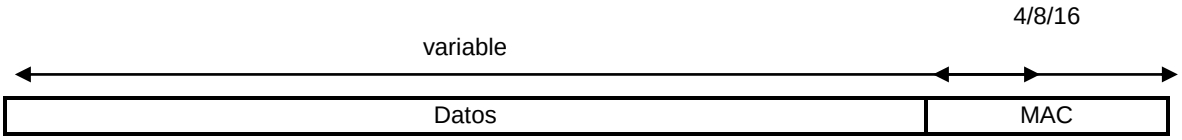

Figura 11. Campo de datos de una trama IEEE 802.15.4 con modo de seguridad CBC-MAC-b, donde b ∈ {4, 8, 16}

CCM. Este usa el modo CCM para cifrado, autenticación e integridad, es decir una combinación de CTR y CBC-MAC. A grandes rasgos, en primer lugar aplica protección de integridad mediante CBC-MAC y luego cifra los datos junto al código MAC empleando el modo CTR. Por tanto, con esta opción, CCM incluye en la trama a transmitir los campos de las operaciones de cifrado y de autentificación: un código MAC, y los contadores de trama y clave.

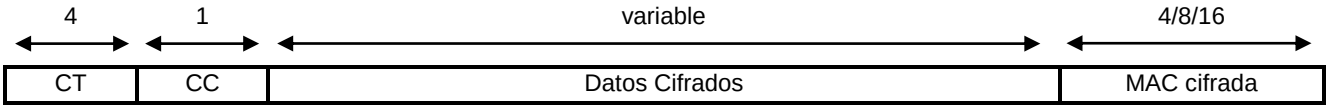

Figura 12. Campo de datos de una trama IEEE 802.15.4 con modo de seguridad CCM-b, donde b ∈ {4, 8, 16}

En la implementación utilizada se toman en cuentas todas estas características para ofrecer los distintos tipos de seguridad.

2.6. AES

Del griego kryptos (ocultar) y grafos (escribir), literalmente escritura oculta, la criptografía es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos.

En la jerga de la criptografía, la información original que debe protegerse se denomina texto plano. El cifrado es el proceso de convertir el texto plano en un galimatías ilegible, denominado texto cifrado o criptograma. Por lo general, la aplicación concreta del algoritmo de cifrado (también llamado cifra) se basa en la existencia de una clave, que es información secreta que adapta el algoritmo de cifrado para cada uso distinto. Las dos técnicas más básicas de cifrado en la criptografía clásica son la sustitución (que supone el cambio de significado de los elementos básicos del mensaje, como letras, dígitos o símbolos) y la trasposición (que supone una reordenación de las mismas); la gran mayoría de las cifras clásicas son combinaciones de estas dos operaciones básicas. El descifrado es el proceso inverso que recupera el texto plano a partir del criptograma y la clave.

El protocolo criptográfico especifica los detalles de cómo se utilizan los algoritmos y las claves (y otras operaciones primitivas) para conseguir el efecto deseado. El conjunto de protocolos, algoritmos de cifrado, procesos de gestión de claves y actuaciones de los usuarios, en su globalidad es lo que constituyen un criptosistema, que es con lo que el usuario final trabaja e interactúa.

En enero de 1997, el Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST), anunció una iniciativa para desarrollar un nuevo estándar de cifrado: El Estándar de Cifrado Avanzado (AES, Advanced Encryption Standard) [4].

En septiembre de 1997, los requerimientos finales para las nominaciones de candidatos para el AES fueron publicados. Los requerimientos funcionales mínimos solicitados para los cifradores simétricos por bloques, incluían la capacidad de ser utilizados con longitudes de bloque de 128 bits y longitudes de clave de 128, 192 y 256 bits.

El ganador fue anunciado en Octubre de 2000, siendo seleccionado como el AES el algoritmo "Rijndael", (el nombre basado en los nombres de los autores, "Rijmen" y "Daemen").

El AES es un cifrador por bloques iterativo, esto es, consiste en la repetida aplicación de una ronda de transformación a la matriz de estado. El número de rondas definidas en el estándar es de 10; este número fue definido por sus diseñadores para el caso de un tamaño de bloque de 128 bits.

Los datos de entrada y salida del AES son consideradas como arreglos de bytes unidimensionales. Para el cifrado, la entrada es un bloque del texto en claro y una clave y la salida es un bloque de texto cifrado. Para el descifrado, la entrada es un texto cifrado y una clave; mientras que la salida es el texto en claro. La entrada corresponde a un bloque del texto en claro de 128 bits y a una clave de cifrado de la misma longitud. La salida corresponde también a un bloque del texto cifrado de 128 bits [4].

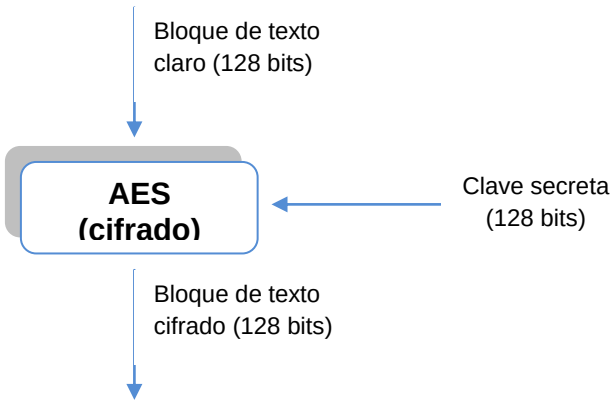

Figura 13. Diagrama de bloques de AES (cifrado) [10]**.**

Cada ronda de transformación del AES, y los pasos que la conforman, operan con una matriz temporal, denominada matriz de estado de 4 × 4. Así mismo, la clave es organizada en una matriz de la misma dimensión.

El cifrado con el AES comienza con la suma inicial de la clave, denominada AddRound-Key, seguida de la aplicación de 9 rondas de transformación y finalmente la ejecución de la FinalRound. La ronda inicial y cada una de las rondas siguientes toman como entrada la matriz de estado y una clave de ronda. La clave de ronda para la *i*-ésima ronda se denota como ExpandedKey { *i* } y ExpandedKey { 0 } denota la clave de entrada para la ronda inicial. La obtención de ExpandedKey de la clave de cifrado se denomina KeyExpansion.

La ronda de transformación se compone de 4 pasos: SubBytes, ShiftRows, MixColumns y AddRoundKey, la ronda final no incluye la aplicación de MixColumns.

Las operaciones básicas aplicadas al bloque son:

- *ByteSub*: aplicando un S-Box (sustituyendo cada byte con otro, basado en una ecuación en GF(2^8));
- *ShiftRow*: El paso de ShiftRows es una transposición de bytes que realiza corrimientos circulares con diferentes desplazamientos a los renglones de la matriz de estado. El renglón 0 es desplazado C_0 bytes, el 1 C_1 bytes, el renglón 2 C_2 bytes y el 3 C_3 bytes, por lo que cada byte en la posición j en el renglón i se desplaza a la posición $(j - C_i) \bmod 4$.
- *MixColumn*: El paso MixColumns es una permutación basada en bloques sobre la matriz de estado realizada columna a columna, la cual puede ser descrita como una multiplicación de la matriz de estado por una matriz constante.
- *Round Key Addition*: Esta ronda consiste en la combinación de la clave de ronda con la matriz de estado mediante la operación lógica XOR.

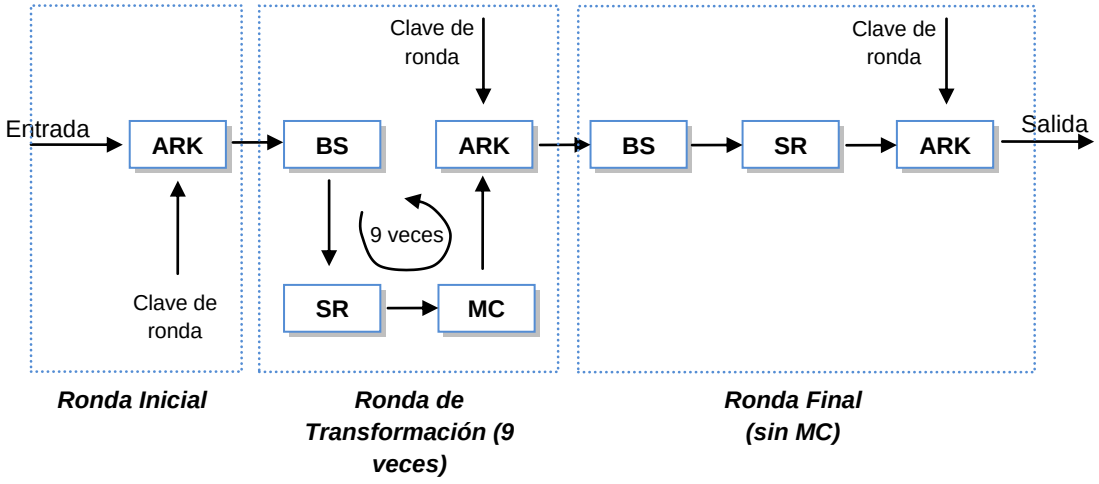

Figura 14. Proceso de cifrado en AES [10]**.**

2.7. Seguridad existente

Existen, como se ha mencionado anteriormente, muchas limitantes en las redes inalámbricas de sensores que deben ser tomadas en cuenta al momento de elegir algún algoritmo criptográfico que ayude a protegerlas de ciertos ataques al momento de compartir información. La memoria y el tiempo de vida de las baterías de las motes, son algunas de estas restricciones a tomar en cuenta, ya que constituyen las principales preocupaciones en contraste con las redes de otro tipo.

Muchos algoritmos han sido implementados y probados para tratar de dar solución a los ataques a la seguridad de la información que se maneja en las redes de sensores, tales como TinySec [11], CBC-X [12], PKI, RSA, SKIPJACK [13], Diffie-Hellman, Discrete Logarithm Problem (DLP), Diffie-Hellman based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), RC5 [14], etc.

Para poder eliminar todas las limitantes que estos algoritmos presentan, se ha hecho uso del modulo de cifrado que ofrece el chip Chipcon CC2420, con el cual trabajan los mote TelosB. Que reduce consumo de potencia y costos de memoria, además de incremento de velocidades al estar implementado en hardware.

3. DEFINICION DEL PROBLEMA A RESOLVER

En la actualidad son muchos los mecanismos de seguridad que el 802.15.4 permite a las redes inalámbricas de sensores. Sin embargo estos mecanismos consumen recursos como memoria, batería, entre otros. Además de incrementar los retardos en la comunicación.

Por lo cual, en el presente trabajo se desea conocer de manera práctica el impacto que la seguridad tiene en el desempeño de este tipo de redes. Para ello se hará una comparación del desempeño de manera teórica, con los valores óptimos apegados a lo especificado en el estándar IEEE 802.15.4.

De manera real se hará uso del sistema operativo TinyOS y de las operaciones de seguridad MAC ofrecidas por el chip CC2420 usado en los motes TelosB.

Para ellos se uso la implementación de JeongGil Ko, estudiante de doctorado y miembro del grupo de investigación Hopkins InterNetworking Research Group (HiNRG) [15], liderado por el Dr. Terzis, que realiza investigación en el ámbito general de las redes informáticas. Los dos principales temas en los cuales están realizando investigaciones son redes de sensores y la seguridad en ellas. Trabajan en múltiples aspectos de las redes de sensores, desde el desarrollo de aplicaciones, diseño de protocolo de red, el procesamiento de la información y gestión de datos.

Esto nos permitirá hacer la comparación con el caso teórico presentado en [3] donde se hace un modelo analítico del impacto de la seguridad en el desempeño de IEEE 802.15.4 dependiendo del modo de seguridad aplicado.

Además veremos el desgaste de la batería, el cual es otro punto importante que se desea conservar en los sensores. Se quiere comprobar el impacto de los mecanismos de seguridad en el tiempo de vida de la batería.

4. METODOLOGIA PARA RESOLVER EL PROBLEMA

4.1. Teórico

Para partir de una base, se han tomado en cuenta las ecuaciones presentadas en [16], con ellas se harán los cálculos de la parte teórica o esperada de las pruebas.

Para calcular el throughput efectivo en cada caso, se consideraron las ecuaciones siguientes:

(1) $T = 8x / delay$

Donde *x* es el número de bytes recibidos (solo de datos de usuario).

El *delay* se define como el retardo total debido a la transmisión de una trama de datos x a una capa superior. Esto se calcula con la siguiente ecuación:

(2) $delay\ (x) = T_{BO} + T_{frame}\ (x) + T_{RT} + T_{ACK} + T_{IFS}\ (x)$

Donde:

T_{BO} = Tiempo de Backoff.
$T_{frame}\ (x)$ = Tiempo de transmisión de la trama.
T_{RT} = Tiempo de roundtrip.
T_{ACK} = Tiempo de transmisión del ACK.
$T_{IFS}\ (x)$ = Tiempos IFS.

En caso de estar usando transmisiones sin el uso de ACK, T_{RT} y T_{ACK} serán omitidos.

En cuanto a los tiempos IFS, dependerán del tamaño de la trama que se esté transmitiendo, pues en caso de tener un tamaño menor o igual a 18 bytes se usara un tiempo corto (SIFS), de lo contrario se considerara el tiempo largo (LIFS). Esto definido por el estándar como ya se mencionó en un apartado anterior.

Para calcular los tiempos de backoff se debe considerar lo siguiente:

(3) $T_{BO} = BO_{slots} * T_{BOslots}$

Donde $T_{BOslots}$ es la duración del slot de backoff, es cual está definido por el estándar como 20 símbolos, y cada símbolo tiene una duración de 16 µseg. Y BO_{slots} es el número de slots de backoff, el cual es un número variable y aleatorio, el cual está dado por una distribución uniforme dentro del rango (0, 2^{BE} – 1), siendo BE la potencia del backoff, y que puede tomar un número mínimo de 3 y un máximo de 5. Por lo cual se pueden tomar valores desde 7 hasta 31.

El tiempo de transmisión de la trama se puede obtener de la siguiente manera:

(4) $T_{frame}(x) = 8 * (L_{PHY} + L_{MAC\_HDR} + L_{addr} + x + \mathbf{Lsec} + L_{MAC\_FTR}) / R$

Donde:

L_{PHY} = Longitud de la cabecera de la capa PHY.
L_{MAC_HDR} = Longitud de la cabecera de la capa MAC.
L_{addr} = Es el campo de longitud de las direcciones utilizadas.
x = bytes enviados,
Lsec = son los campos de control usados por el mecanismo de seguridad utilizado.
L_{MAC_FTR} = Longitud del footer de la capa MAC.
R = Es la velocidad nominal de transmisión de datos.

En el caso de usar ACK en los envíos de información, tendrá que considerarse la siguiente ecuación:

(5) $T_{ACK} = 8 * (L_{PHY} + L_{MAC\_HDR} + L_{MAC\_FTR}) / R$

Estas son las ecuaciones que han sido tomadas como base para obtener los resultados teóricos, a estos se les han dado los valores óptimos basándose en el estándar, por lo cual los resultados obtenidos de ellas son los valores que se esperaban bajo el impacto de estos mecanismos. Estos resultados se verán en contraste con el caso práctico más adelante.

4.2. Práctico

Para la realización de los pruebas prácticas, se realizó un pequeño programa que realiza envíos de datos de un sensor a otro, uno programado como estación base o receptor y otro como simple emisor, tomando como base una ambiente ideal, es decir solo una fuente y un receptor, por lo cual no había colisiones.

En la realización de las pruebas se tuvo que considerar el tamaño máximo de datos de usuario en cada caso. Ya que en caso de superar el payload permitido (127 bytes) los sensores no realizan ninguna operación.

JeongGil Ko realizó la implementación de seguridad CC2420 soportada para TinyOS 2, haciendo posible la evaluación de la red con respecto a su desempeño bajo CTR (cifrado / descifrado), CBC – MAC (autenticación e integridad) y CCM (integridad, autenticación y cifrado / descifrado). En el caso de CBC y CCM pudiendo usarse distintos tamaños para el código MAC (4, 8 ó 16).

Esta implementación permite elegir entre los tres modos de seguridad, en TinyOS la función fue llamada SecAMSend y provee una interface que proporciona estas opciones, mediante los comandos:

- call CC2420Security.setCtr(a, b);
- call CC2420Security.setCbcMac(a, b, c);
- call CC2420Security.setCcm(a, b, c);

Ellas habilitan la seguridad seleccionada antes del envío de cada paquete. Cada una de ellas maneja tres parámetros, que identifico como *a, b* o *c*, para su explicación.

El primero ***a,*** permite al usuario seleccionar la clave, recordando que hay espacio en memoria para dos claves.

El segundo parámetro ***b*****,** establece el número de bytes del payload que se desea no se tomen en cuenta para el cifrado (en el caso de CTR y CCM) y la autenticación (en el caso de CBC y CCM). Por defecto este parámetro es 0, ya que lo normal es iniciar luego de las cabeceras y tomar en cuenta todo el payload para iniciar estas operaciones.

El tercer parámetro ***c*****,** es usado en CBC-MAC y CCM para especificar la longitud del código MAC (código de autenticación de mensaje) [10]. Los valores pueden ser elegido entre los números 4, 8 y 16.

Luego de indicar los valores respectivos en las funciones, se puede llamar la interfaz de envío AMSend de manera normal.

Otro punto importante a destacar en la implementación, es la adaptación del *nonce* (campos añadidos en la cabecera por los mecanismos de seguridad) dentro de la cabecera, es decir, la especificación de los campos FrameCounter y Key que se pueden observar en el código dentro de CC2420.h mostrado a continuación:

```
typedef nx_struct security_header_t {
 nx_uint8_t secLevel;
 nx_uint8_t keyMode;
 nx_uint8_t reserved;
```

```
 nx_uint32_t frameCounter;
 nx_uint8_t keyID[1];
} security_header_t;
```

Con esta implementación se hizo la comparación entre el desempeño de los sensores de manera teórica (valores deseados) con las ecuaciones que se presentaron en el apartado anterior, en contraste con los resultados que se obtuvieron en TinyOS cuando se usan de manera real estos mecanismos de seguridad.

Para el apoyo a esto se usó el software llamado Perytons-M, desarrollado por la empresa Perytons [17] ubicada en Israel, que consiste en una herramienta de monitoreo y análisis para Redes Inalámbricas de Sensores 802.15.4. Este programa permite analizar todas las tramas 802.15.4 que se encuentren cerca del ordenador que posee el software. Al ordenador se le debe conectar una antena especial llamada *dongle* la cual es la que recibe la trama y la envía al software.

Su interfaz gráfica muy amigable, además cuenta con características que ningún otro analizador posee. Entre dichas características se pueden encontrar las siguientes:

- ✧ Análisis multiredes: 802.15.4 define 16 canales posibles de comunicación en la banda ISM 2.4 GHz. El analizador Perytons puede capturar múltiples canales simultáneamente permitiendo analizar procesos dinámicos como pueden ser el escaneo activo, múltiples redes operando en el mismo espacio o verificar que los dispositivos no estén transmitiendo en canales no asignados.

- ✧ Confiabilidad: Este analizador provee alta confiabilidad en la recepción utilizando técnicas de diversidad en las antenas para reducir el número de pérdidas de paquetes, ocurridas generalmente por los efectos de interferencias en la propagación. Con este software se pierden menos del 0.1 % de las tramas según los datos del fabricante.

- ✧ Sofisticación: Para uso profesional, el analizador posee un conjunto de herramientas que permiten identificar escenarios problemáticos. Los mensajes pueden ser buscados por valor, texto o existencia de un cierto campo o jerarquía de éstos. Los mensajes pueden ser fácilmente comparados para así encontrar las diferencias o similitudes según se desee. También pueden ser organizados de acuerdo a una gran variedad de opciones.

- ✧ Fácil de usar: El uso de interfaces gráficas, hace que el uso de las funcionalidades del analizador sea muy sencillo e intuitivo con un proceso de entrenamiento muy corto. La ayuda que trae el software es bastante clara y pocos problemas pueden surgir en un análisis sencillo.

- ✧ Enfoques: Este analizador permite ver la captura desde el punto de vista de tiempo o de mensaje. En la primera, se muestra en una línea de tiempo los

mensajes que han llegado, y en la segunda muestra todos los mensajes que han llegado con su correspondiente información.

Por último se hará uso del Agilent Technologies DC Power Analyzer N6705A, un analizador de potencia que nos permitirá ver el desgaste de las baterías en cada uno de los casos, es decir, cuando los sensores están haciendo envíos simples sin seguridad, y cuando están usando alguno de los mecanismos de seguridad implementados (CTR, CBC, CCM).

5. RESULTADOS EXPERIMENTALES

En este apartado se irán presentando los distintos resultados de las pruebas realizadas. Con el uso de sensores TelosB, el sistema TinyOS y el analizador Perytons, todos mencionados en apartados anteriores.

Primero se comprobó que las cabeceras añadidas dependiendo del mecanismo fueron los correctos. En la Tabla 2 se puede ver la cantidad del payload ***n*** (datos enviados) y la longitud final de la trama según el mecanismos implementado.

Tabla 2. Tamaño de trama, dependiendo de mecanismo utilizado en bytes.

n	NO SEC	CTR	CBC-MAC-4	CBC-MAC-8	CBC-MAC-16	CCM4	CCM8	CCM16
60	74	79	78	82	90	83	87	95
65	79	84	83	87	95	88	92	100
70	84	89	88	92	100	93	97	105
75	89	94	93	97	105	98	102	110
80	94	99	98	102	110	103	107	115
85	99	104	103	107	115	108	112	120
90	104	109	108	112	120	113	117	125
95	109	114	113	117	125	118	122	130
100	114	119	118	122	130	123	127	135
105	119	124	123	127	135	128	132	140
110	124	129	128	132	140	133	137	145
115	129	134	133	137	145	138	142	150

Con los datos de la Tabla 2 podemos ver cuál es la longitud máxima de bytes de datos de usuario que pueden ser enviados dependiendo del tipo de seguridad que se desea implementar, pues se debe recordar que el tamaño máximo de la trama (incluyendo cabeceras) no podrá ser mayor a 127 bytes. En la tabla aparecen en color rojo aquellas tramas que sobrepasan esta regla. De esta manera podemos darnos cuenta que mientras Sin Seguridad pueden enviarse 113 bytes de datos de usuario, en el caso de CCM16 el máximo sería de tan solo 92 bytes.

En la Figura 15 se puede ver como incrementan las tramas, solo tomando en cuenta los mecanismos que más overhead añaden, pues la diferencia entre los otros mecanismos es mínimo. Como ya se comprobó en la Tabla 2.

Entre una trama Sin seguridad y una usando CCM-16, hay una diferencia de 21 bytes, que son los bytes de contador de trama, clave y los 16 de MAC de autenticidad (que se especifica al hacer la llamada a la función). Esto se pudo comprobar visualmente con el analizador de protocolo, pues en él se muestra las tramas tal y como son enviadas.

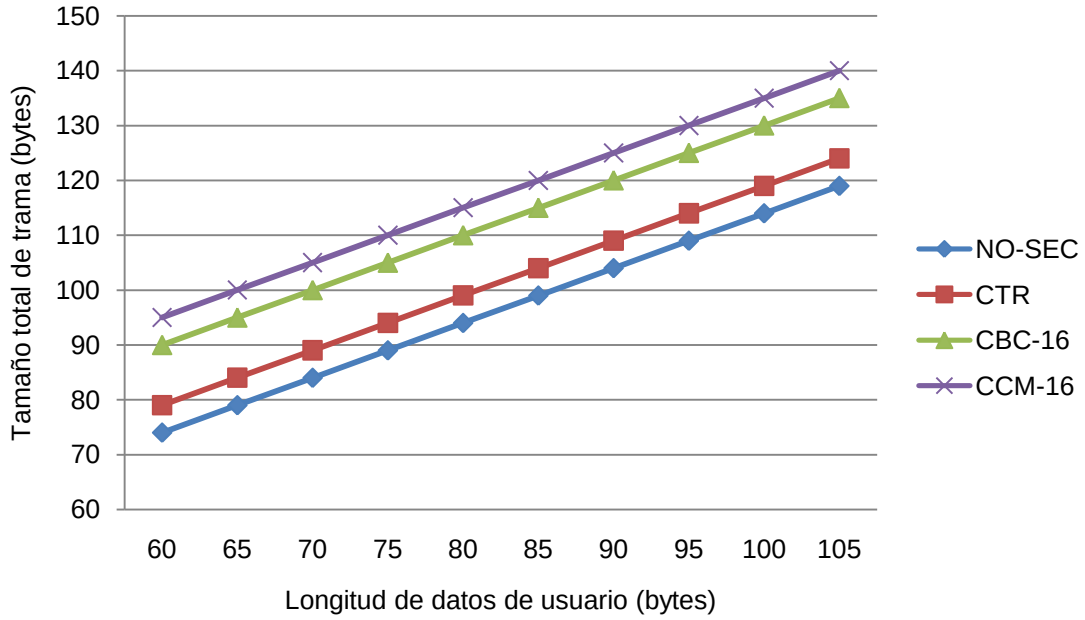

Figura 15. Tamaño de trama, dependiendo mecanismo de seguridad.

En las figuras siguientes se puede observar cómo se presentan las tramas en el analizador.

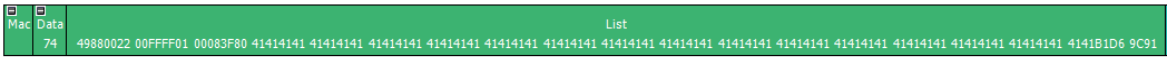

Figura 16. Trama sin seguridad

En la Figura 16 se muestra el PSDU (PHY Service Data Unit) de una trama de 60 bytes sin seguridad, en ella podemos observar cada uno de las partes que la conforman.

FCF (Frame Control Field): 4988
PAN Id: 0022
No. De secuencia: 00
Destino: FFFF
Fuente: 0100
Type= 08
6LowPAN: 3F (para poder interactuar con estas redes)
Length: 80 (en decimal, corresponde a 128, que es el tamaño máximo)
DATA= 41414141414141414141414141414141……
FCS: 9C91

Esto definido por el sistema TinyOS en el archivo CC2420.h como se muestra en el código siguiente:

```
typedef nx_struct cc2420_header_t {
        nxle_uint16_t fcf;
        nxle_uint16_t destpan;
        nxle_uint8_t dsn;
        nxle_uint16_t dest;
        nxle_uint16_t src;

        /** I-Frame 6LowPAN interoperability byte */
        #ifdef CC2420_IFRAME_TYPE
                nxle_uint8_t network;
        #endif

        #ifndef TINYOS_IP
                nxle_uint8_t type;
        #endif
}cc2420_header_t;
```

En el caso de una trama cifrada, la información enviada vía radio es capturada por el analizador de esta manera:

Figura 17. Trama con mecanismos de seguridad CTR.

Como se puede observar, el envío está protegido y no se puede interpretar la información dentro de la trama.

En el caso de la información que solo es autenticada, la trama se ve de la siguiente manera:

Figura 18. Trama con mecanismo de seguridad CBC-MAC-4, en hexadecimal

En esta figura 18 se puede observar la información enviada, que para el ejemplo hemos usado letras A, que el analizador muestra en formato hexadecimal (41). Además de poder observar al final de los datos, el código MAC generado que protege la trama.

En los campos de cabecera se puede ver también como se activa un Flag cuando lo que se envía ha sido enviado bajo algún mecanismo de seguridad. Esto se hace modificando el control de registro de seguridad de la radio.

Estos registros son SECCTRL0 y SECCTRL1. Y visualmente se ve como en la Figura 19.

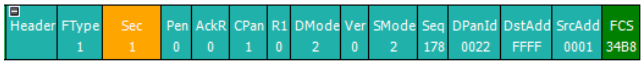

Figura 19. Flag de seguridad activado.

A partir de estos datos, los cuales se visualizan en el analizador, podemos darnos cuenta como está estructurada la información que viaja vía radio y confirmar que el cifrado se llevo a cabo, además de saber el tiempo que se tarda en enviar estas tramas.

5.1. Impacto de la Seguridad en los Retardos de Transmisión

Se hizo la programación de las motes, éstas hacían el envío de mil tramas con diferentes longitudes de datos de usuario, desde 60 bytes hasta el máximo soportado por cada mecanismo de seguridad. Esto nos daba el tiempo total del envío de estas tramas y a partir de ahí se pudo promediar el tiempo en milisegundos para cada una de ellas.

Esos tiempos se resumen en la Tabla 3:

Tabla 3. Retardos de los envíos (en mseg)

Bytes	No-SEC	CTR	CBC-MAC-4	CBC-MAC-8	CBC-MAC-16	CCM-4	CCM-8	CCM-16
60	9,31	10,03	9,85	10,17	10,81	10,36	10,67	11,30
65	10,00	10,46	10,36	10,67	11,29	10,81	11,17	11,79
70	10,45	10,81	10,74	11,08	11,77	11,17	11,47	12,18
75	10,80	11,23	11,14	11,48	12,19	11,57	11,98	12,59
80	11,21	11,67	11,60	11,52	12,60	12,03	12,32	13,00
85	11,64	12,14	11,95	12,32	13,01	12,46	12,82	13,44
90	12,14	12,63	12,45	12,80	13,42	12,97	13,31	13,97

En estos resultados podemos ver que la diferencia entre enviar una trama sin cifrar y una trama cifrada (CTR) es 0.72 milisegundos. Con respecto a una autenticada (CBC MAC-16) es de 1.5 milisegundos. Y con respecto a una trama cifrada y autenticada (CCM16), la diferencia incrementa a 1,99 milisegundos.

A partir de esta información y en base a las ecuaciones planteadas para el caso teórico, las cuales están fundamentadas con el protocolo IEEE 802.15.4, se llegó a obtener resultados sobre el throughput efectivo en cada uno de los casos.

La ecuación utilizada es T = 8x / delay, donde *x* fue la longitud de datos de usuario que se enviaba en ese momento (entre 60 y 90 bytes). Los retardos son los

presentados en la Tabla 3, los cuales fueron obtenidos como se mencionó anteriormente con el analizador de protocolos.

Como se puede observar en las Figuras 20 y 21, la distribución en cuanto a resultados es bastante parecida, más no los rangos obtenidos. Es decir que se puede observar claramente la diferencia entre una trama sin ningún tipo de seguridad, la cual no tiene overhead extra ni procesos añadidos al envío. Y aquella a la cual se le ha implementado algún tipo de seguridad permitido.

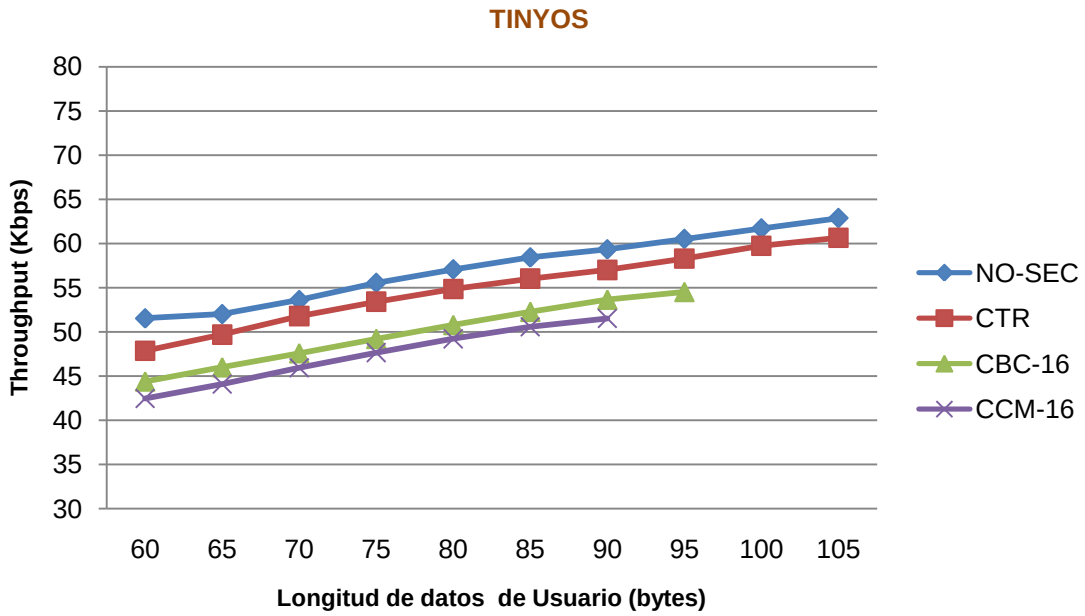

Figura 20. Throughput efectivo, caso práctico (TinyOS). NO ACK.

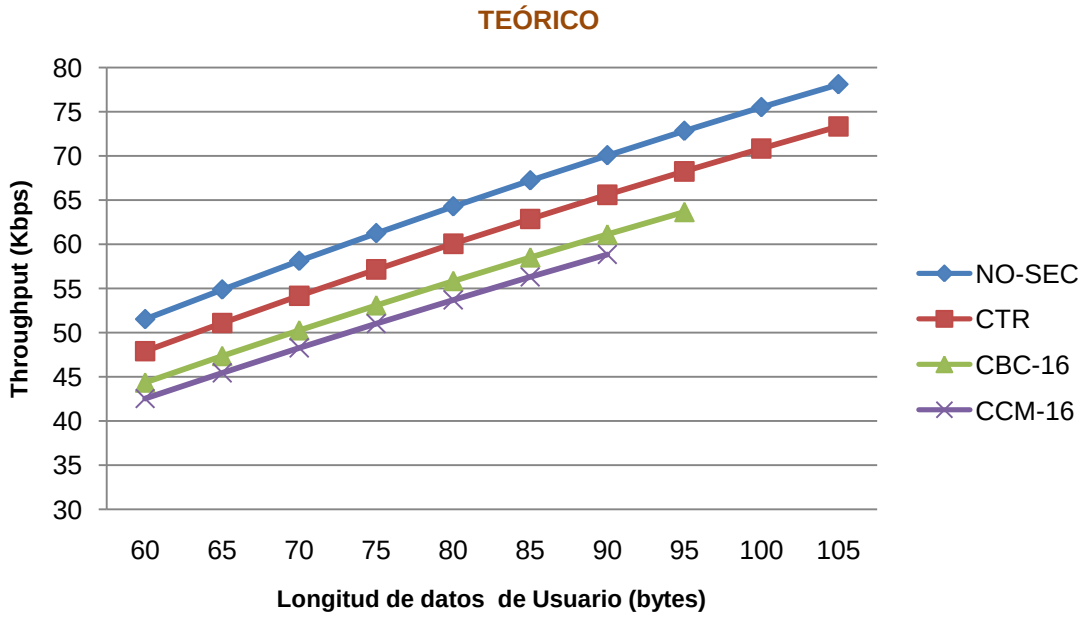

Figura 21. Throughput efectivo, caso teórico. NO ACK.

Pero en cuanto a los rangos de throughput se puede observar una diferencia de alrededor del 7 y 16% entre el caso teórico y el caso práctico. Esto está dado por los tiempos de backoff. Es decir, en el caso teórico se considera una backoff fijo independiente de los envíos, de los tamaños de trama y de cualquier otro factor que se presentara. Al realizar las pruebas reales no se presento de esta manera. Ya que TinyOS está programado para que los backoff sean tan aleatorios como sean posibles lo cual hace que los backoff no sean fijos. Provocando que el throughput tenga un decremento considerable.

Esto se presentó durante las pruebas, ya que al realizar los cálculos del throughput se observaron las discrepancias, lo cual se estuvo analizando hasta que observamos que los tiempos de backoff entre trama y trama eran largos y aleatorios durante todo el proceso de transmisión. En el caso teórico, se plantea un backoff inicial y fijo durante todo el proceso, lo cual no se presentó en las pruebas reales.

En el caso de las pruebas prácticas los backoff que se obtuvieron fueron entre 6,31 y 9.33 milisegundos.

Otras pruebas fueron realizadas con el uso de ACK en los envíos. Esto dio como resultado los tiempos de la Tabla 4.

Tabla 4. Retardos de los envíos usando ACK (en mseg)

Bytes	No-SEC	CTR	CBC-MAC-4	CBC-MAC-8	CBC-MAC-16	CCM-4	CCM-8	CCM-16
60	10,30	12,65	12,59	12,94	13,54	12,99	13,39	14,02
65	10,68	13,05	12,96	13,33	14,02	13,40	13,76	14,42
70	11,14	13,46	13,39	13,75	14,41	13,50	14,20	14,87
75	11,53	13,91	13,80	14,21	14,81	14,22	14,56	15,23
80	11,97	14,41	14,23	14,56	15,29	14,72	15,05	15,69
85	12,43	14,88	14,74	15,06	15,69	15,22	15,54	16,18
90	12,92	15,28	15,17	15,49	16,13	15,63	15,89	16,51

En este caso, los resultados en cuanto a throughput obtenido se muestran en las Figuras 22 y 23. Donde podemos ver la misma línea que en el caso anterior, esto significa, el mismo comportamiento pero con una diferencia entre el 4 y 12 % en cuanto al que es posible alcanzar de manera teórica. Todo también dado por los tiempos de backoff entre tramas, que al igual que en el caso anterior, se presentan más extensos y aleatorios que como se preveían en el caso teórico.

Estos tiempos se presentaron desde un mínimo de 6.94 milisegundos a un máximo de 11,52 milisegundos. En el caso teórico se tomaba el mínimo como un fijo constante en todos los envíos. Debido a esto se produjo el descenso en el throughput obtenido en el caso real frente al caso teórico.

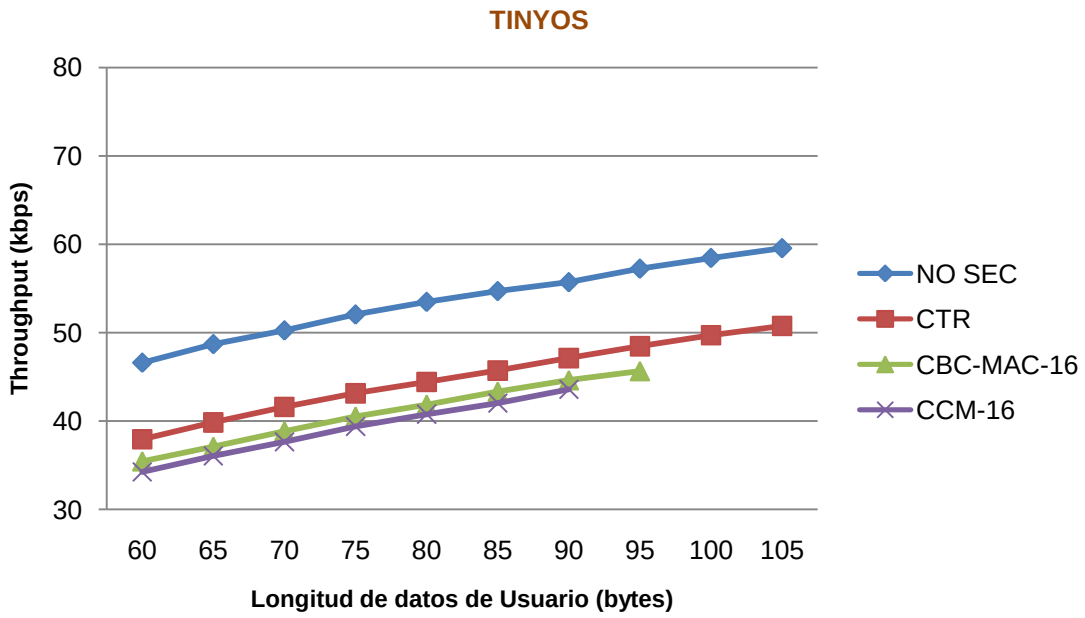

Figura 22. Throughput efectivo, caso práctico (TinyOS). CON ACK.

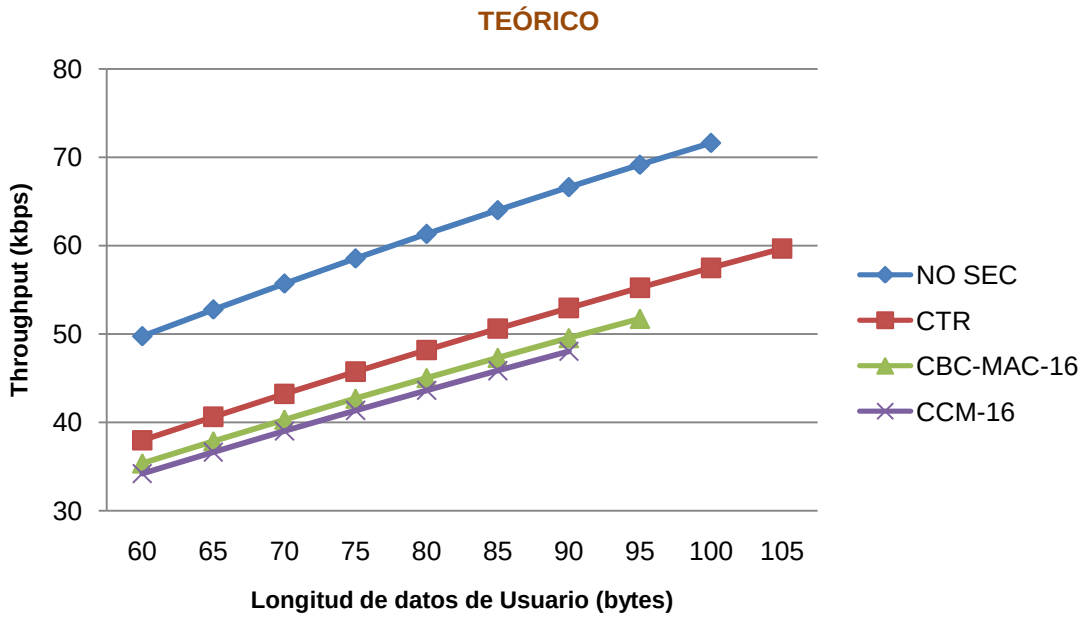

Figura 23. Throughput efectivo, caso teórico. CON ACK.

Con esto se puede ver que de igual manera que en el caso de la omisión de los ACK, el comportamiento es similar a lo esperado de manera teórica, alto Throughput para aquellos envíos que no están implementando nada extra en su procedimiento, y un descenso en aquellos que hacen una generación de código MAC de 16 bytes, además de hacer el cifrado de toda la información, añadiendo un overhead de 21

bytes en total. Pero esta baja de Throughput además se ve afectada por los backoff aleatorios que el TinyOS tiene implementados en su programación.

Después de mucho análisis sobre este efecto, sobre por qué el backoff no se mantiene constante, se repaso el código de las funciones de TinyOS donde se encuentra implementado, se hicieron algunas modificaciones sin obtener cambio en los resultados. Por lo cual se llegó a la conclusión de que estos tiempos están implementados en el chip de transmisión CC2420 y no se pudieron modificar los algoritmos que lo controlan.

Por este motivo se presentan las diferencias entre los valores del throughput en el caso teórico en relación al práctico. Ya que para obtenerlo se tomó como base:

- $T = 8x / delay$
- $delay\ (x) = T_{BO} + T_{frame}\ (x)$

x, es la longitud de datos de usuario, el cual se utilizó desde 60 bytes hasta 105 en los casos en que fue posible.

El tiempo de transmisión de trama se obtuvo a partir de lo siguiente:

- (8 * Trama / *R*), donde la Trama incluye las cabeceras, y *R* es 250 Kbps.

El *delay* o retardo estuvo dado por el analizador, como se mostró en la Tabla 4.

A partir de ahí se pudo obtener cual era el backoff que se aplicaba en cada envío y así poder obtener los throughput. El resumen de todos estos datos se muestran en la Tabla 5 que se me presenta a continuación.

Tabla 5. Resumen de datos para envíos sin seguridad

Bytes de usuario	Transmisión de Trama (mseg)	Retardo Total (mseg)	Backoff TinyOS (mseg)	Throughput TinyOS (Kbps)	Throughput Teórico (Kbps)
60	2,37	9,31	6,31	51.54	51,51
65	2,53	10,00	6,83	52.01	54,86
70	2,69	10,45	7,12	53.60	58,10
75	2,85	10,80	7,31	55.54	61,24
80	3,01	11,21	7,57	57.07	64,27
85	3,17	11,64	7,83	58.42	67,21
90	3,33	12,14	8,17	59.33	70,05

Como se puede ver, en TinyOS los promedios de backoff son variables, dependiendo los tamaños del la longitud de datos de usuario manejados. Lo cual al final afecta con una disminución en el throughput que se obtiene, porque mientras en el teórico se tomarían 6.31 milisegundos como inicial y fijo constante durante todo el proceso de transmisión, como se muestra en la tabla, en el caso práctico este fue en aumento desde 6.31 hasta 8.17 milisegundos.

Esto se presenta gráficamente a continuación, donde se ve como inician con el mismo throughput pero conforme se aumenta la longitud de datos de usuario en TinyOS va en aumento el backoff y por consiguiente el incremento en el throughput no es tan notorio como en el caso teórico que mantuvo un backoff fijo.

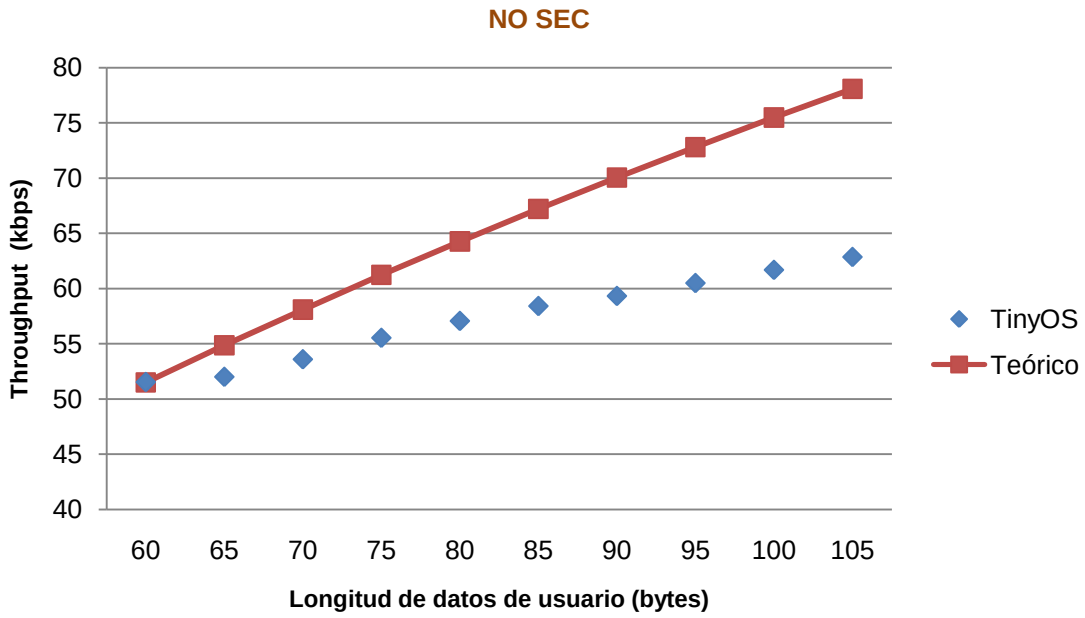

Figura 24. Validación de resultados sin seguridad.

5.2. Impacto de la Seguridad en el Consumo de Energía

En el caso de la energía requerida, se hizo uso de un analizador de potencia para poder observar el comportamiento de consumo de la carga al momento que los sensores están transmitiendo o recibiendo información. Esto, haciendo diferentes lecturas, primero se inicio haciendo pruebas cuando los sensores trabajan sin ningún tipo de seguridad, en decir de forma normal solo enviado datos a una estación base.

Este analizador funciona como una fuente de alimentación de energía, que permite la medición de voltaje e intensidad. Esto se hace conectado los sensores al analizador sustituyendo la alimentación de las baterías convencionales. Este nos permite visualizar los consumos mediante graficas o guardando las datos obtenidos en ficheros para su posterior análisis.

Mediante su panel se hace la configuración de carga y voltaje requeridos, y en la pantalla se visualizan los resultados. En la Figura 25 se muestra el analizador así como la manera en que se realizó la conexión de los sensores a él.

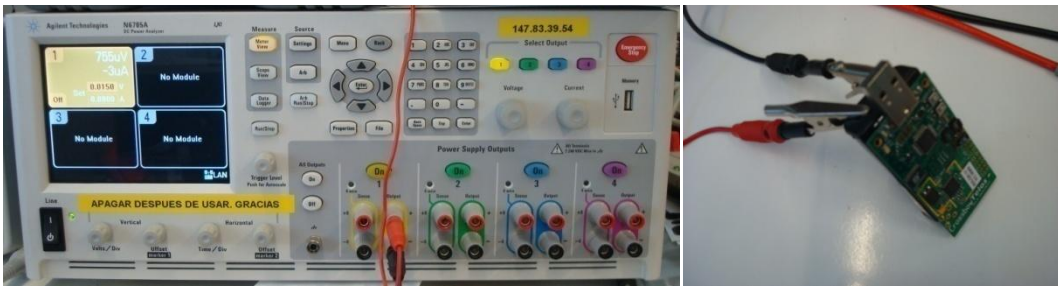

Figura 25. Analizador sustituyendo fuente de alimentación en un mote.

Las mediciones se hicieron tanto en el nodo transmisor, como aquel programado como Estación Base y que recibe toda la información que el emisor está generando.

Se realizaron pruebas con las diferentes llamadas a las funciones de seguridad, de igual manera usando un nodo emisor y una Estación Base. Llegando a la conclusión de realizar pruebas donde en el mismo programa se hicieran envíos de la misma cantidad de paquetes sin seguridad, con CCM16 y después pasar a estado de reposo, esto para ver el cambio bajo el mismo entorno.

Se realizó una prueba con los motes programados para envíos de tramas de 90 bytes de usuario sin seguridad, pasando a la misma cantidad de envíos con la misma longitud de datos de usuario pero bajo la implementación de CCM16 y se finalizaba con un tiempo en reposo, es decir, sin actividad alguna.

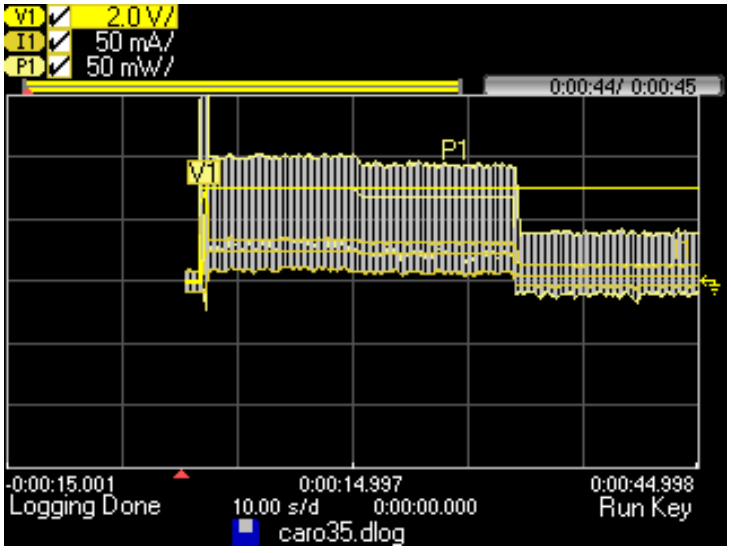

Figura 26. Resultado del analizador de potencia en Transmisión.

El resultado de lo anterior se pudo observar en el analizador como lo muestra la Figura 26, donde el cambio más notorio es cuando pasa de transmitir a estado de reposo el nodo que hace los envíos.

Como resultado de las lecturas con el analizador se pudo obtener que como media, la corriente usada para transmitir paquetes sin cifrar es de 24 mA, y en recepción 26 mA. En el caso del envío de datos bajo la seguridad CCM16 la media en los envíos es de 22 mA y de igual manera que en el caso anterior 26mA en recepción. Esto sin hacer uso de ACK y aplicando un voltaje de 3 V. Esto en la gráfica que presenta el analizador no es tan notorio puesto que la escala es pequeña. En la Figura 27, se puede distinguir claramente como en los envíos sin seguridad hay una carga de 24 mA, cuando pasa a CCM16 hay una disminución de 2 mA y en el caso de reposo baja hasta 4.7 mA, en el caso de transmisión.

Estos resultados se deben a que en el caso de no presentar el añadido de ningún método de cifrado o seguridad, la carga es constante y el envío se hace con mayor rapidez, requiriendo una energía menor que en el caso contrario.

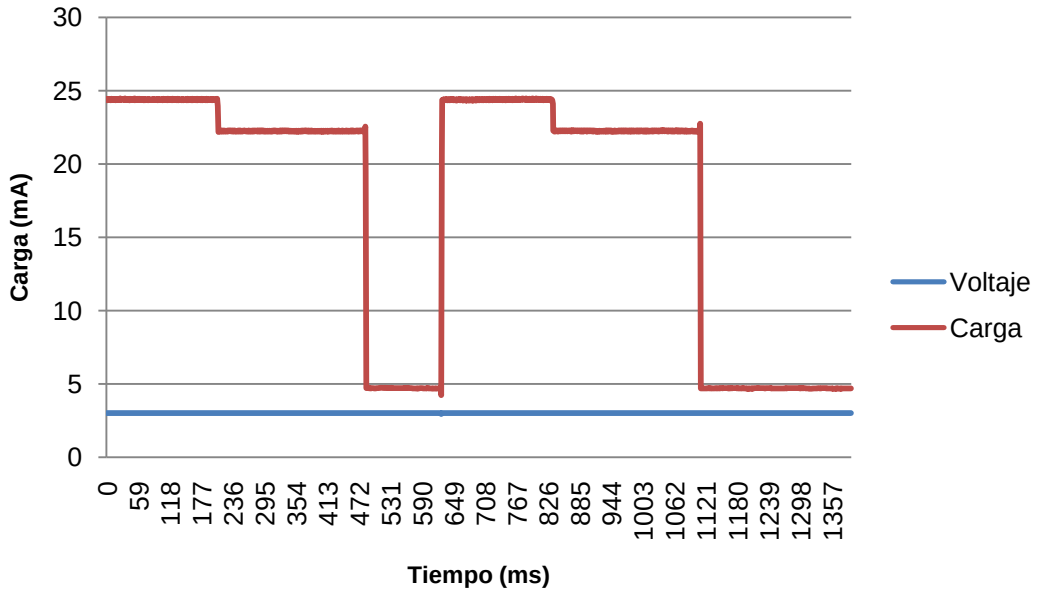

Figura 27. Transmisión de paquetes Sin Seguridad y CCM16.

En el caso de recepción, mientras no recibe ningún paquete se encuentra en un nivel de carga de 24 mA en promedio. Esto porque siempre está en estado de espera, es decir en un constante monitoreo de recibir información, por ello su descenso no es mayor. Esto se puede observar claramente en la Figura 28, en la cual se observa constante la recepción de datos en 26 mA y el pequeño descenso en caso de no recibir nada.

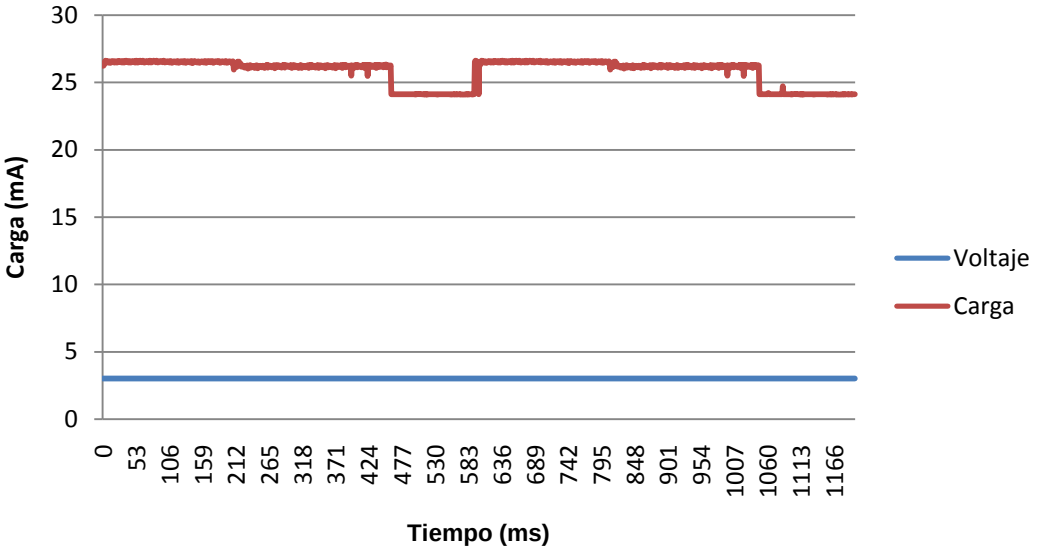

Figura 28. Recepción de paquetes Sin Seguridad y CCM16.

En la Tabla 6 se resumen el desgaste de energía, el cual depende de los tiempo de envío que se necesita para cada caso, esto lo obtuvimos anteriormente con el analizador.

Tabla 6. Resumen de energía requerida para diferentes envíos.

	NOSEC			CCM16		
bytes	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**
60	0,0093	0,6696	0,7254	0,0113	0,7458	0,8814
65	0,0100	0,72	0,78	0,0118	0,7788	0,9204
70	0,0104	0,7488	0,8112	0,0122	0,8052	0,9516
75	0,0108	0,7776	0,8424	0,0126	0,8316	0,9828
80	0,0112	0,8064	0,8736	0,0130	0,858	1,014
85	0,0116	0,8352	0,9048	0,0134	0,8844	1,0452
90	0,0121	0,8712	0,9438	0,0140	0,924	1,092
95	0,0126	0,9072	0,9828	--	--	--
100	0,0130	0,936	1,014	--	--	--
105	0,0134	0,9648	1,0452	--	--	--

En la Figura 29 podemos ver como la recepción con seguridad CCM16 gasta más energía, esto se debe a que en caso de implementar un tipo de seguridad los tiempos de envíos y recepción son mayores.

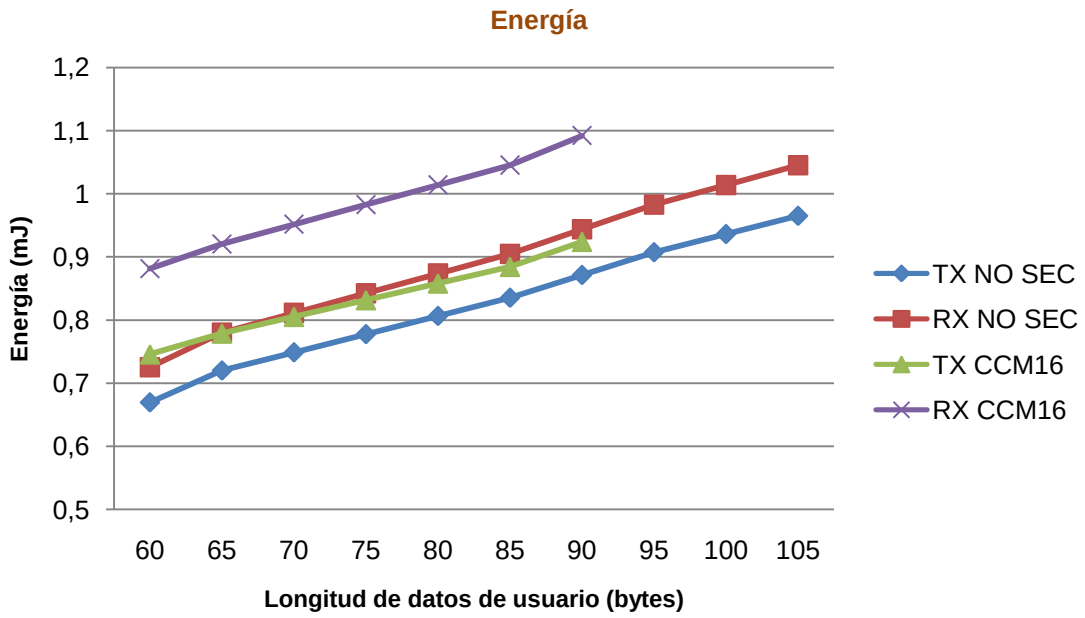

Figura 29. Energía necesaria tanto en transmisión como en recepción.

Una batería convencional usada en los motes, tiene una capacidad de 2000mA·hora, y tomando en cuenta los datos obtenidos anteriormente en cuando a la transmisión de tramas y cargas correspondientes, se puede llegar a la conclusión de que la cantidad de envíos dependiendo de la longitud de datos de usuario.

Esto es multiplicando el tiempo de transmisión por la carga necesaria, en este caso 22 o 24 mA según corresponda. El valor obtenido esta dado en mA·seg, entonces a continuación se pasa a mA·Hora y se divide entre los 2000 que soporta la batería. Esto se resume en la Tabla 6.

Tabla 7. Cantidad de tramas transmitidas con una batería AA

	Tramas ($x\ 10^6$)	
bytes	**NOSEC**	**CCM16**
60	116	98
65	109	93
70	103	89
75	97	85
80	92	81
85	87	78
90	83	75

6. CONCLUSIÓN

Durante el desarrollo de este proyecto se llegaron a varias conclusiones en cuanto a la seguridad en las redes inalámbricas y principalmente en las redes de sensores IEEE 802.15.4.

Entre las principales es que se deben tomar en cuenta todas las restricciones en este tipo de redes de comunicación al momento de proporcionar seguridad en ellas, puesto que sus características son muy diferentes a las redes inalámbricas tradicionales, por lo cual se deben tener consideraciones especiales al momento de implementar la seguridad, como el desgaste de la batería, el bajo alcance, poca memoria disponible, etc.

Aún con estas restricciones no es imposible la implementación de seguridad; como se mencionó en el presente proyecto, existen varias maneras de ofrecer seguridad dependiendo el tipo de sensor a utilizar, ya que no todos soportan las mismas funcionalidades.

En el caso de los motes TelosB utilizados en este proyecto, se pudo comprobar que el overhead añadido por la implementación de seguridad del chip CC2420 es el correcto, con un máximo de 21 bytes añadidos. Esto haciendo que cada tipo de implementación provoque distintos retardos, afectado esto en el throughput efectivo en los envíos.

También se pudo conocer que la programación de los backoff afecta en los resultados esperados, puesto que de manera teórica se tomaron valores fijos que en la práctica no están programados de esa manera, lo cual provocó diferencias en los resultados.

Por último se comprobó la carga que necesitan los sensores para los envíos, tanto en recepción como en transmisión. En este punto se ha podido observar claramente que en el caso del uso de seguridad tiene un mayor desgaste de energía, lo cual generó un mayor consumo de batería. Esto relacionado con los tiempos de envío necesarios.

7. LÍNEAS FUTURAS

En el transcurso de la elaboración de este proyecto, se han podido localizar varias líneas para nuevas investigaciones relacionadas con este tema. Entre ellas las que se pueden destacar principalmente son:

- ✧ Realizar las pruebas prácticas presentadas en este proyecto, pero bajo la influencias de un protocolo de multisalto, es decir que ahora para que el emisor llega al correspondiente receptor tenga que pasar antes por más de de una mote.

- ✧ Modificar la programación del sistema operativo TinyOS o del chip CC2420 en la parte de protocolo de acceso al medio para tratar de reducir los backoff, ya que al reducirlos provocara reducción en los retardos y aumento de throughput.

- ✧ Por último, otra área que es factible para su estudio es tratar de hacer uso de otro tipo de seguridad.

8. REFERENCIAS

[1]. M., Healy, T., Newe y E., Lewis. *Efficiently securing data on a wireless sensor network.* s.l. : SENSORS07, 2007.

[2]. IEEE, Computer Society. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).* New York : s.n., 2006.

[3]. Gomez, Carlos, Casademont, Jordi y Paradells, Josep. *Theoritical Study on the Impact of Security Mechanisms on Performance of IEEE 802.15.4 and ZigBee higher layers.* Barcelona : s.n., 2008.

[4]. Rodríguez Henríquez, F., y otros. *Cryptographic Algorithms on Reconfigurable Hardware.* Berlin : Springuer, 2006. ISBN: 978-0-387-33883-5.

[5]. Crossbow. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf. [En línea]

[6]. Portal TinyOS Comunity Forum. [En línea] http://www.tinyos.net/.

[7]. Installing TinyOS 2.0.2. [En línea] http://www.tinyos.net/tinyos-2.x/doc/html/install-tinyos.html.

[8]. David Gay, David Culler, Philip Levis. http://www.tinyos.net/api/nesc/doc/ref.pdf. [En línea] 2002.

[9]. Chipcon. http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf. [En línea] 2006.

[10]. López Trejo, Emmanuel. *Implementación Eficiente en FPGA del Modo CCM usando AES.* México, DF : s.n., 2005.

[11]. Karlof, Chris, Sastry, Naveen y Wagner, David. *TinySec: a link layer security architecture for wireless sensor networks.* s.l. : Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004.

[12]. Li, Shiqun, y otros. *Eficient Link Layer Security Scheme for Wireless Sensor.* s.l. : Journal of Information and Computational Science, Binary Information Press, 2007.

[13]. NIST. [En línea] 29 de Mayo de 1998. http://jya.com/skipjack-spec.htm. Version 2.0.

[14]. R., Rivest. *The RC5 Encryption Algorithm.* MIT Laboratory for Computer Science : s.n., 1997.

[15]. Hopkins interNetworking Research Group. [En línea] http://hinrg.cs.jhu.edu/.

[16]. Latre, B. *Throughput and Delay Analisys of Unslotted IEEE 802.15.4.* 2006.

[17]. Perytons Network Visibility. [En línea] http://www.perytons.com/.

a. LISTA DE ACRÓNIMOS

ACK	ACKnowledgments
AES	Advanced Encryption Standard
ASK	Amplitude Shift Keying
BPSK	Binary Phase-Shift Keying
CAP	Contention Access Period
CBC	Cipher Block Chaining
CCA	Clear Channel Assessment
CCM	Counter Mode CBC MAC
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CTR	Counter Mode
DLP	Discrete Logarithm Problem
ECDLP	Elliptic Curve Discrete Logarithm Problem
FCF	Frame Control Field
FCS	Frame Check Sequence
FIFO	First In – First Out
GTS	Guaranteed Time Slot
IEEE	Institute of Electrical and Electronics Engineers
IFS	InterFrame Space or Spacing
ISM	Industrial, Scientific, and Medical
LIFS	Long InterFrame Spacing
LQI	Link Quality Indication
LR-WPAN	Low Rate Wireless Personal Area Networks
MAC	Medium Access Control

MAC	Message Authentication Code
MFR	MAC Footer
MHR	MAC Header
MPDU	MAC Protocol Data Unit
NesC	Network Embedded Systems – C
NIST	National Institute of Standards and Technology
O-QPSK	Offset Quadrature Phase Shift Keying
PAN	Personal Area Network
PHR	PHY Header
PHY	Physical Layer
PKI	Public Key Infrastructure
PPDU	PHY Protocol Data Unit
PSDU	PHY Service Data Unit
RC5	Cifrado por Bloques, Rivest Cipher
RSA	Sistema criptográfico con clave pública. (Rivest, Shamir y Adleman)
SHR	Synchronization Header
SIFS	Short InterFrame Spacing
WSN	Wireless Sensor Metwork