Congreso Internacional en Ingeniería Electrónica. Mem. Electro 2015, Vol. 37, pp 92-97, Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2015
ISSN 1405-2172

DESARROLLO E IMPLEMENTACIÓN DE UNA ESTACIÓN BASE PARA UNA RED INALÁMBRICA DE SENSORES DE LARGO ALCANCE CON CONEXIÓN A LA NUBE.

César Ortega-Corral[1], Oscar R. Acosta Del Campo[1], Jesús Enrique López Montoya[1], José Jaime Esqueda Elizondo[2], Luis E. Palafox[2], Ricardo Guerra Frausto[2], Roberto A. Reyes[2], Florencio López Cruz[1]

[1]Tecnologías de la Información y Comunicación
Universidad Tecnológica de Tijuana
Km. 10 Carretera Libre Tijuana-Tecate,
Fracc. El Refugio. Quintas Campestre. Tijuana, B.C., C.P. 22650
Tel +52 (664) 969 9700 . Ext. 4789
Correo-e: cesar.ortegac@uttjiuana.edu.mx , oscar.acosta@uttijuana.edu.mx, jesus.lopez@uttijuana.edu.mx

[2]Facultad de Ciencias Químicas e Ingeniería.
Universidad Autónoma de Baja California.
Calzada Tecnológico 14418, Mesa de Otay, Tijuana, B.C., C.P. 22390
+52 (664) 682 1033 – 5800
Correo-e: jjesqueda@uabc.edu.mx,

RESUMEN

En este trabajo se presenta la integración de una estación base (BS, del inglés *Base Station*) de alto rendimiento, encargada en recibir telemetría de una red inalámbrica de sensores (WSN, *Wireless Sensor Network*) con arquitectura jerárquica de largo alcance. El diseño incorpora un sistema procesador de 32 bits ARM (*Advanced RISC Machine*) en comunicación serie con un transceptor digital de 900MHz (con opción a operar a 2.4GHz), a través del cual la BS recibe los mensajes inalámbricos que envían los nodos de la WSN. Para el envío de los mensajes de los sensores hacia la nube de datos, la estación base establece comunicación mediante una interfaz Ethernet con conexión a un *Gateway* vía direccionamiento IP (Internet Protocol). Dichos mensajes están codificados usando un protocolo modificado denominado Light JSON (Java Script Object Notation); por lo que la estación base con programación orientada a objetos se encarga de traducirlos a JSON estándar y de enviarlos a un servidor remoto a través de Sockets TCP/IP. La codificación y decodificación sirve como parte del proceso de la validación de la información en tránsito. Estos mensajes que viajan de extremo a extremo de la WSN, contienen datos de aplicación y métricas de otras capas de la red inalámbrica, que se agregan mientras realizan los saltos establecidos en el enlace, hasta un servidor dedicado donde se guarda la información en un sistema de base de datos (DBS, *data-base system*). Aquí se presenta la actualización del hardware de la estación base, la migración de su software, sus funciones, la ampliación de sus operaciones y el mejoramiento en el manejo del paradigma de validación con acuses de recibo.

Palabras clave: WSN, JSON, Estación Base inalámbrica.

ABSTRACT

We present a high performance base station (BS) development, in charge of receiving telemetry from a hierarchical long-range Wireless Sensor Network (WSN). Our BS design incorporates a serially connected 32-bit ARM (Advanced RISC Machine) processor system to a configurable 900MHz transceiver (with the option of 2.4 GHz operation), through which the BS receives wireless messages sent by the WSN nodes. For data transfer from the wireless sensors towards the data cloud, the BS uses its Ethernet interface connected to the gateway to establish communications using IP (Internet Protocol) addressing. The wireless sensor data messages are coded using a modified protocol called Light JSON (Java Script Object Notation). The base stations object oriented software is in charge of decoding incoming LJSON messages and of converting them into standard JSON, and of sending them to a remote server through TCP/IP Sockets. The decoding and coding processes serve as in transit information validation procedures. These end-to-end WSN messages contain application data and metrics from other wireless network layers that are aggregated while they hop through the wireless links, until they get to the dedicated server where the relevant information is stored within a database system (DBS). Here we present the updated base stations hardware, its functions, its augmented operations and the enhanced validation paradigm with acknowledgment receipt messages.

Keywords: WSN, JSON, TCP, Wireless Base Station.

1. INTRODUCCIÓN

En la actualidad, mucho se ha publicado sobre los nodos que componen los diversos tipos de WSN, tales como: nodos terminales (EP, *end-points*), nodos enrutadores y nodos de cabecera de clústeres (CLH, *cluster-heads*) [1,2]. Pero aún es poca la cantidad de publicaciones sobre el nodo *sink*, y en consecuencia poco publicado sobre estaciones base (BS, *base-station*); lo que hay se limita a decir que las BS reciben mensajes de los nodos inalámbricos identificados por su dirección MAC, y que su radio *sink* transmite anuncios en *broadcast*, por lo que en el mensaje es donde se indica el direccionamiento, y es donde se envían palabras de sincronía y control [3]. También, en la literatura se escribe que los nodos

sink se conectan a un sistema digital con mayores recursos, el cual debe estar programado para recibir los mensajes de todos los nodos distribuidos en la WSN [4]. Así mismo, las BS que se han desarrollado como parte de plataformas experimentales, son nodos inalámbricos básicos conectados por USB (*universal serial bus*) a una Laptop o computadora de escritorio; por lo que suele usarse dicho sistema como servidor en sitio, con la posibilidad de emplear una simple DBS (*data-base system*) o de almacenar los datos en archivos de texto plano con formato CSV (*coma seperated values*) [5]. Aunque se ha buscado mejorar el servicio con el desarrollo del llamado *middle-ware* de una WSN operando en la computadora de la BS, donde se suelen aplicar arquitecturas orientadas a servicios (SOA, *service oriented applications*) en el mejor de los casos [6,7]; pero para cuando se usa SOA se supone que ya se habrán almacenado los datos captados por el *gateway* de la WSN a través de algún *back-end* de muy bajo nivel, sistema programado que puede ser poco robusto si no se le da importancia en el aspecto de validación de los datos. Es decir, no hay mucho publicado en dos aspectos: (1) sobre el hardware que pudiera estar completamente dedicado a funcionar como estación base; y también es insipiente lo publicado (2) sobre el software en el servidor encargado de recibir y validar los mensajes de la WSN, así como en filtrar y almacenar los datos en el último del sistema con amplios recursos. Por necesidad, y aprovechando el nicho de oportunidad, aquí se presenta la integración y codificación de una BS, dedicada en coordinar una WSN, tanto en su operación, captura y transferencia de datos. Y se completa la comunicación de extremo a extremo, del lado de un servicio PHP/JSON operando como *back-end*, para recibir los mensajes JSON traducidos y codificados por la BS para la validación y almacenamiento final de los datos en una DBS y en archivos de valores separados por comas (o CSV, *comma seperated values*) como respaldo redundante.

2. ARQUITECTURA DE LA WSN

En este desarrollo, la WSN opera en la intemperie en condiciones adversas de humedad y temperatura, cuyos nodos ambientales potencialmente están dispersos a kilómetros de distancia de la estación base [8]. Las aplicaciones de esta WSN jerárquica, son de monitoreo marino como ambiental, su arquitectura se ilustra en la Fig. 1.

Aunque se trata de una red jerárquica, desde el punto de vista de la estación base sólo hay nodos que se asocian a su radio *sink* como destino general, y de cuáles se recibe su telemetría, que puede ser de los nodos que la envían o de otros nodos que se encuentran en otra parte de la WSN. La red jerárquica está conformada por dos niveles de comunicaciones. En el nivel inferior se ubican los nodos EP, que disponen de escasos recursos pero suficientes para hacer el muestreo de sus sensores y el envío de los datos resultantes a un nodo intermediario que controla a los grupos o clústeres de EP, por lo que se le llama *cluster-heads*, CLH. Por lo que en el nivel superior se ubican los CLH en comunicación con la estación base.

3. SELECCIÓN DE TECNOLOGÍAS

En el desarrollo anterior, se puso en operación una estación base para una WSN de monitoreo marino construida alrededor del conocido módulo microcontrolador RCM4300 de 16 bits, operando a 80MHz, dispone de cuatro puertos RS232, mediante uno de ellos se establece la conexión en serie con el radio AC4490, el cual representa el nodo *sink* de la WSN. Una de las cualidades por las que se había seleccionado el RCM4300 es que abordo dispone de un controlador y puerto Ethernet, para establecer conexiones TCP/IP (Transport Control Protocol / Internet Protocol) con un servidor remoto. Otra razón por la que se seleccionó el RCM4300 es que ofrece el paradigma que llaman co-procesos, mediante los cuales se facilita la implementación de máquinas de estados usando su nativo lenguaje Dynamic C. La desventaja en usar dicho

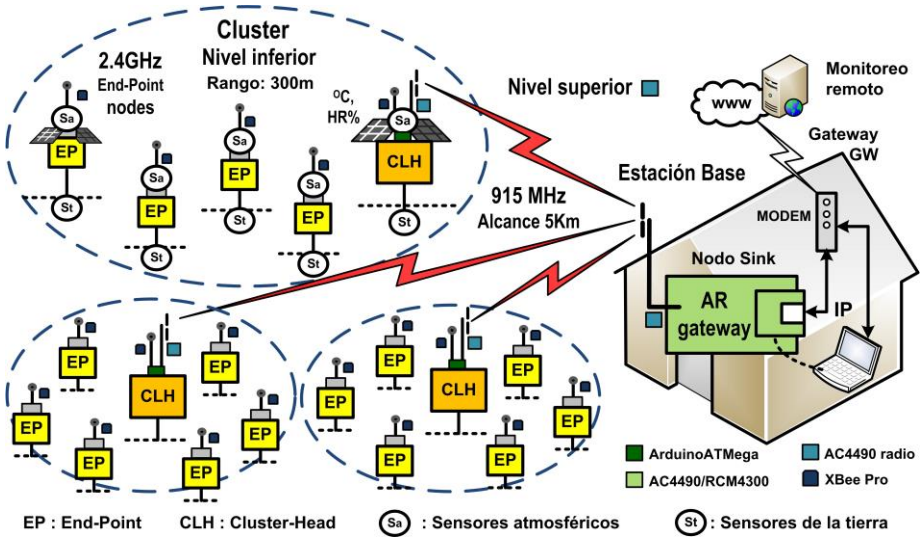

Fig. 1. Arquitectura jerárquica de la WSN para monitoreo ambiental.

módulo es que los co-procesos mal elaborados o excesivos causan inestabilidad de la ejecución de las instrucciones, y en ocasiones el código se pierde en saltos prematuros en el código, dado que el esquema de prioridades provoca que ciertas operaciones del software se ejecuten de modo asíncrono y en des-tiempos indeseables.

Para realizar la integración de la BS en poco tiempo, y sobre todo en el desarrollo de software, se optó por emplear un procesador ARM de 32 bits de hardware abierto conocido como Arduino Due [9], opera a 84MHz y es capaz de ejecutar 104 MIPS (millones de instrucciones por segundo). En la Fig. 2 se muestra en qué parte del sistema opera el Arduino Due como BS. La razón de no usar hardware con mayores recursos, que pudiera incluso contar con sistema operativo e interfaz gráfica avanzada, es que no es necesario si la solución no lo requiere. En este caso, se aprovecha hardware de comunicaciones previamente empleado para integrar nodos CLH Arduino Mega de 900MHz y 2.4GHz, se trata de un tablilla de frecuencia dual con dos conectores, uno para un radio AC4490 de 900MHz [10] y otro para el conocido XBee de 2.4GHz [11].

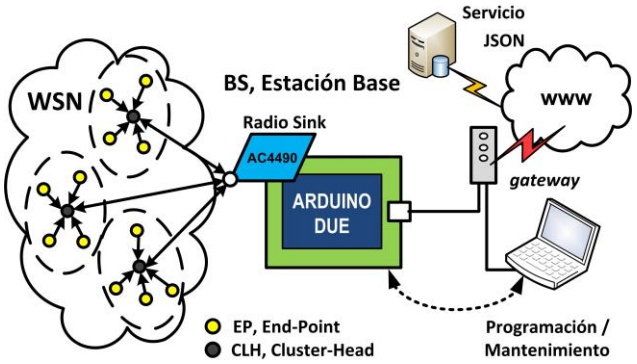

Fig. 2. Uso del Arduino Due como procesador central de la estación base.

4. PROTOCOLOS DE COMUNICACION

La comunicación entre los nodos y la estación base se hace mediante el envío de mensajes que se generan desde la capa de aplicación, los cuales contienen información de las capas inferiores; por ejemplo: de la capa de aplicación se agrega el sello de tiempo, de la capa física se incluyen los voltajes de baterías y de los valores medidos de los sensores, de la capa de enrutamiento se agrega la dirección asignada al nodo origen, y al recibir a nivel enlace de datos, la cabecera de las tramas indican el tipo de mensaje y la dirección MAC del radio origen. Los mensajes de aplicación se codifican usando un protocolo de 8-bits modificado, basado en la sintaxis de la conocida notación de 16 bits llamada JSON (*JavaScript Object Notation*), en este caso para WSN se le ha llamado LJSON (o "light" JSON) [12]. Por lo que la estación base se dedica en convertir los mensajes L-JSON a JSON estándar, y vice-versa, lo cual implica la reasignación de los mensajes de caracteres de 8 bits a que sean de 16 bits, y viceversa; esto incluye la necesidad de agregar caracteres redundantes para cumplir con las reglas de JSON estándar. Un mensaje sencillo LJSON ejemplo se muestra a continuación.

```
{C:1,T:"1430891231",En:1,e:[{i:2,T:
"1430891230",An:4,Av:[235,9E8,81C,430]}],[{An:
5,Av:[126,21F,1C8,110,789]}}
```

A diferencia de JSON estándar que usa comillas para toda literal y valor, en Light JSON se usan comillas sólo para distinguir cadenas de caracteres o sellos de tiempo que son valores numéricos, que exceden la longitud de palabra de valores enteros largos. Todo mensaje LJSON es un objeto que empieza con el caracter "llave que abre", **{**, y termina con "llave que cierra", **}**, pero evita usar extensamente las comillas como lo establece JSON, cuyo valor le precede a los "dos puntos", **:**, como par asociativo **etiqueta:valor**. En el mensaje ejemplo la etiqueta **C** indica qué Cluster-Head lo envió, según convenido previamente, la **T** es el sello de tiempo, la etiqueta **En** indica el número de End-Points que tienen datos agregados al mensaje, esa información la guarda el arreglo **e:[ ]**. En este caso, el único EP que envía datos es el nodo 2, dado por su identificador **i:2**, para el EP se incluye su propio sello de tiempo **T:"1430891230"**, y envía **An=4** muestras contenidas en el arreglo **Av:[,,,]**. Así mismo, el CLH también envía en el mensaje sus propios valores resultantes del muestreo de sus **An=5** sensores, valores que están almacenados en el segundo arreglo **Av:[,,,,]**. La conversión del mensaje LJSON de 8-bits a JSON estándar de 16-bits se realiza haciendo "casting" de las variables en el software y agregando las comillas a las literales, por lo que el ejemplo anterior ahora en JSON sería el siguiente mensaje:

```
{"C":1,"T":"1430891231","En":1,"e":[{"i":2,
"T":"1430891230","An":4,"Av":[235,9E8,81C,430]
}],[{"An":5,"Av":[126,21F,1C8,110,789]}}
```

Ya construido el mensaje JSON, la BS se encarga de abrir una conexión IP usando Sockets para comunicarse con el Servicio JSON y transferirle el mensaje codificado.

5. ALGORITMOS DE LA ESTACIÓN BASE

Los algoritmos de la BS se codificaron en una plataforma de programación C++ orientada a objetos llamada Processing para *Arduino Due*. El paradigma se explica mediante los casos de uso descritos por los diagramas de transición que se muestran en la Fig. 3, donde se presentan tres operaciones fundamentales realizadas: (1) la BS anuncia el tiempo actual para que se actualicen todos los relojes de tiempo real de los nodos *cluster-heads* (CLH1, CLH2, ..., CLHN); (2) la BS recibe en TDM (*time division multiplexing*) los mensajes de cada CLH, donde vienen agregados los datos del muestreo automático de los EP, donde la BS actúa como faro emitiendo un mensaje de sincronía cada 30 segundos; (3) la BS establece

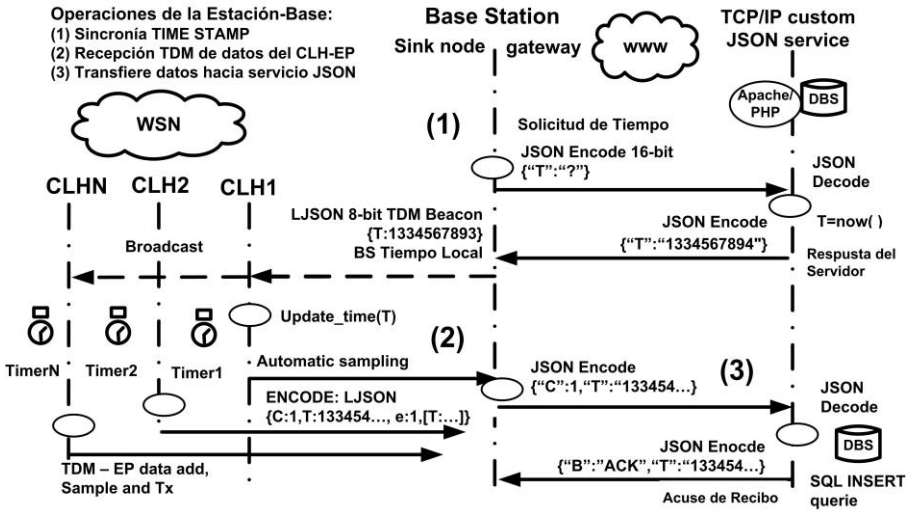

Fig. 3. Operaciones básicas de la BS de (1) sincronía de tiempo, (2) muestreo y envío de datos en TDM, y (3) codificación JSON y transferencia a un servicio en la nube del Internet.

salida de los datos de la WSN hacia la nube, operando como un decodificador/codificador LJSON a JSON, y como *gateway* realizando el envío de datos hacia el Servicio JSON.

6. TRANSFERENCIA DE DATOS HACIA LA NUBE.

El programa del servidor está formado por un conjunto de algoritmos, que realizan diversas funciones tales como: (1) Establecer conexiones TCP/IP escuchando por un Socket, (2) realizar la validación de los mensajes JSON, pasándolos por un filtro que verifica la sintaxis, (3) y que extrae de manera confiable los datos contenidos en los mensajes decodificados, y (4) almacena en un DBS la información recibida de los sensores remotos y de las métricas de la calidad de los enlaces. En la Fig. 4 se desglosan las fases particulares del muestreo y envío automático de mensajes LJSON, desde los CLH hasta la BS, y de los mensajes JSON de la BS hacia el Servidor TCP/IP. Los CLH emiten mensajes por turnos una vez que reciben la señalización (o *beacon*) TDM de la BS, operando como faro emitiendo un mensaje en *broadcast* cada 30 segundos.

7. RESULTADOS

Para probar el desempeño de la BS basada en Arduino Due, se instaló el sistema en la intemperie junto con dos CLH, colocados a corta distancia de la BS, usando atenuadores balanceados de RF (radio-frecuencia) en la conexión de su antena, se emplearon atenuadores de -20dB marca Crystek que operan desde 0Hz hasta 3GHz [13], con lo cual se emulan mayores distancias. En este caso lo que se probó fue la recepción de los mensajes LJSON multiplexados en el canal inalámbrico. En la Fig. 5 se presenta el monitor serial que despliega el inicio del proceso de la BS. Por otro lado, en la Fig. 6 se muestra la consola del servidor JSON, cuando recibe de la BS la solicitud del sello de tiempo UNIX, esta es la comunicación inicial de BS con el Servidor, tal como se mostró en la Fig. 3 anterior. En la Fig. 7, se muestra un ejemplo de los datos almacenados por el servidor JSON, representa la curva de temperatura medida por el CLH-1 durante 62 horas. Se observan claramente las temperaturas máximas de día y mínimas de noche. Registros similares se obtuvieron del segundo CLH, puesto en operación para probar

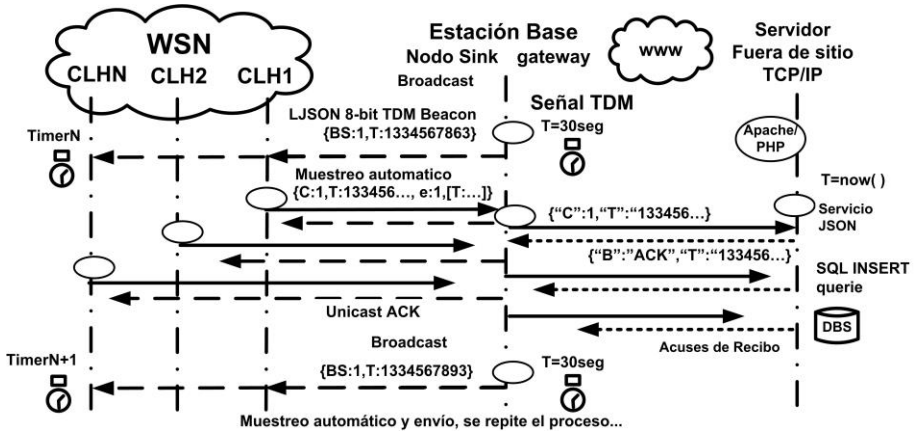

Fig. 4. Operaciones básicas de envío de datos TDM desde los CLH hacia la BS, el proceso es cíclico.

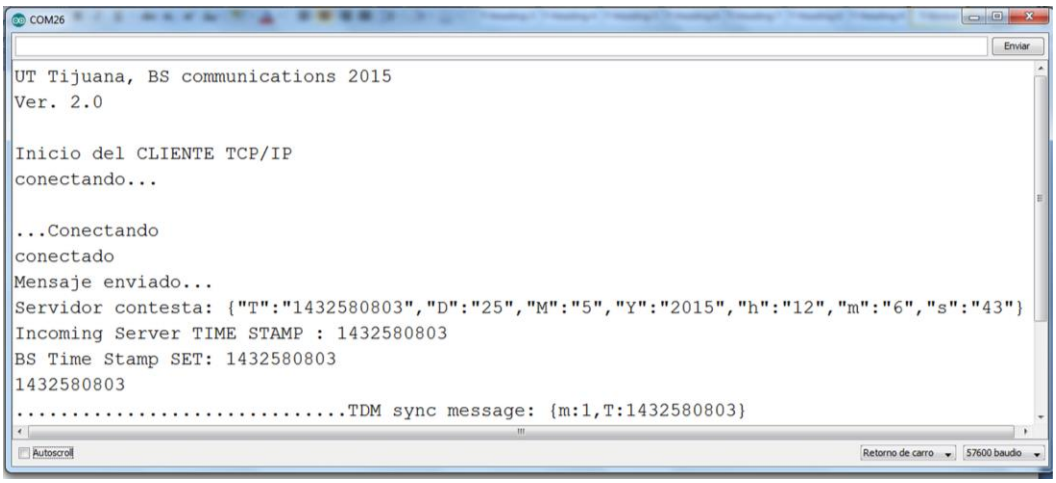

Fig. 5. Monitor serial de la BS, mientras inicia su operación solicitando y recibiendo el sello de tiempo.

el paradigma actualizado del software que opera en la estación base.

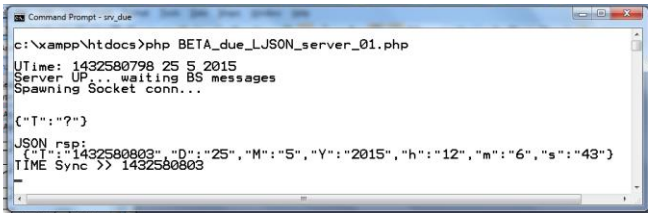

Fig. 6. Consola del servidor JSON, muestra el mensaje que recibe la BS que da inicio a la comunicación mediante Sockets.

Por otro lado, en la Fig. 8 se presenta la curva de carga y descarga de baterías del CLH-1 empleando un panel solar y cargador adecuado, donde también se observa el comportamiento de carga máxima día y de descarga de noche.

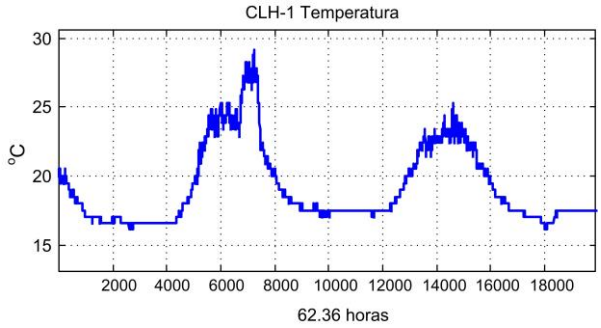

Fig. 7. Mediciones de temperatura del CLH1, datos almacenados en la base de datos por el Servicio JSON.

Los valores de otras variables, como humedad relativa y presión atmosférica, también fueron adquiridas por los CLH y enviados a la BS en mensajes LJSON. La BS al recibirlos, convirtió los mensajes a JSON estándar, abrió el socket correspondiente para luego enviar los mensajes al Servicio JSON encargado de extraer los datos, validarlos y almacenarlos en la base de datos.

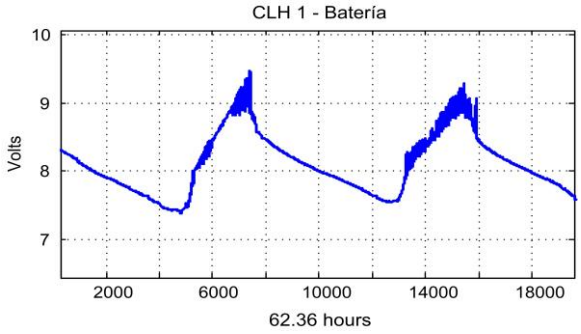

Fig. 8. Curvas de carga de la batería del CLH1, alimentada por un panel solar.

8. CONCLUSIONES

Típicamente, en redes inalámbricas de sensores, los datos adquiridos por los nodos son enviados al nodo *sink*, para luego entregarlos a la estación base. Y aunque hay estándares sobre la formación de las redes, no hay consenso sobre la arquitectura y operación de estaciones base para WSN. Por lo que hay un nicho de oportunidad en la integración de BS. El requisito primordial para una BS es que debe ser un sistema de alto rendimiento, capaz de recibir mensajes de múltiples orígenes. En algunos casos, se han dispuesto computadoras portátiles como estaciones base que a su vez operan como servidores, con la capacidad de almacenar la información en bases de datos. Pero ese paradigma se queda corto en cuanto a portabilidad y en que puedan tener presencia los datos en la nube de Internet. Aquí se actualizó un paradigma que propone una estación base centrada en un sistema ARM, con interfaz Ethernet para el envío de los datos a una servidor remoto, que incluye mecanismos de validación de los datos antes de su almacenamiento.

REFERENCIAS.

[1] Z. H. Khan, et. al. ***Hierarchical Wireless Network Architecture for Distributed Applications***. IEEE 5th Conference on Wireless and Mobile Communications, 2009. pp. 70 - 75. 23 - 29, august 2009.

[2] Winston Seah and Yen Kheng Tan. ***Sustainable Wireless Sensor Networks***. Hard cover book, 574 pages. Publisher: InTech. December 2010.

[3] Suchita R.Wankhade1 and Nekita A.Chavhan. ***A review on data collection method with sink node in Wireless Sensor Networks***. International Journal of Distributed and Parallel Systems (IJDPS) Vol.4, No.1, January 2013.

[4] Hidehiro Nakano, et. al. ***A Sink Node Allocation Scheme in Wireless Sensor Networks Using Suppression Particle Swarm Optimization***. Publisghed in Sustainable Wireless Sensor Networks journal, chapter 17, Intech publishing. January 2010. ISBN 978-953-307-297-5

[5] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, Jean Marie Bonnin. ***Wireless sensor networks: a survey on recent developments and potential synergies.*** The Journal of Supercomputing April 2014, Volume 68, Issue 1, pp 1-48.

[6] Herrera-Quintero, L.F., et. al. ***Wireless Sensor Networks and Service-Oriented Architecture, as suitable approaches to be applied into ITS***. 6th IEEE Euro American Conference on Telematics and Information Systems (EATIS), 23-25 may 2012.

[7] Leguay, J.; Lopez-Ramos, M. ; Jean-Marie, K. ; Conan, V.. ***An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks.*** 33rd IEEE Conference on Local Computer Networks, 2008. pp 740-747. Montreal, Canada. 2008.

[8] César Ortega-Corral et. al. ***End-to-end message exchange in a deployable marine environment hierarchical wireless sensor network***. International journal of distributed sensor networks. Vol. 2014, Article ID 950973. ISSN 1550-1329.

[9] Arduino Due. **http://www.arduino.cc/en/Main/ArduinoBoardDue Sitio** visitado el 28/04/2015.

[10] Laird Tech. ***AC4490 900MHz Transceiver***. User's Manual. Ver. 3.2.1., 2007.

[11] Digi Inc. ***XBee™/XBee–PRO™ 2.4 GHz OEM RF Modules***. Product Manual v1.xCx – 802.15.4 Protocol For OEM RF Module Part Numbers: XB24–...–001, XBP24–...–001 IEEE® 802.15.4 OEM RF Modules by Digi International. MaxStream. Digi Inc. 2008. USA.

[12] C. Ortega-Corral, L.E. Palafox, J.A. García-Macías, L. Aguilar, J. Sánchez-García, A. C. Valenzuela-León, I. Chon-Aguiar. (2012) ¬ ***A 'Lighter' JSON for message exchange in highly resource constrained wireless sensor network applications***. Congreso Internacional de Electrónica ELECTRO 2012, Chihuahua. Mexico. Oct. 2012

[13] Crystek Corp. ***http://www.crystek.com/home/attenuator/atten.aspx*** Sitio web consultado el 10/06/2015.