Implementación de Red de Internet de las Cosas Usando OpenMote y Raspberry Pi

Matías Gastón Casteluccio, Sacha Agustín Niemetz,
Juan Carlos Taffernaberry y Rodrigo Gonzalez

GridTICs, Universidad Tecnológica Nacional, Rodriguez 273, Mendoza, M5502AJE, Argentina.
casteluccio.m@gmail.com, saniemetz@hotmail.com, {cartaffe, rodralez}@frm.utn.edu.ar

Resumen Las redes de Internet de las Cosas (IoT) han ido ganando popularidad en los últimos años, debido a un amplio campo de aplicación y uso. En el presente trabajo se propone una mejora respecto a un desarrollo anterior basado en la Computadora Industrial Abierta Argentina (CIAA). Se propone utilizar una computadora embebida Raspberry Pi para centralizar la implementación de diferentes componentes del sistema IoT. Los resultados de las pruebas llevadas a cabo en laboratorio y en campo demostraron la correcta operación del sistema IoT propuesto.

Keywords: IoT · Raspberry Pi · 6LoWPAN · CoAP.

1. Introducción

La comunicación autónoma entre diferentes elementos sin la participación de los seres humano se conoce hoy como la Internet de las Cosas (IoT, por sus siglas en inglés)[1]. Estos son dispositivos electrónicos embebidos dentro de objetos de uso cotidianos que permiten su conexión a Internet, con el objetivo de lograr el acceso remoto a los datos generados, o el accionamiento de actuadores a distancia. Son implementados generalmente con microcontroladores con bajo poder de procesamiento, poseen comunicación inalámbrica y sensores que permiten obtener valores de parámetros del medio ambiente tales como temperatura, humedad, posición, entre otros. Estos objetos normalmente están interconectados y forman redes de área personal (PAN, por sus siglas en inglés).

Las redes de área personal pueden ser cableadas o inalámbricas. En este último caso se denominan redes inalámbricas de área personal (WPAN), siendo el estándar IEEE 802.15 el más utilizado. Además, la IEEE desarrolló el estándar IEEE 802.15.4 [2] para ser aplicado a redes inalámbricas de baja potencia y baja tasa de transferencia, características típicas de las WPANs, denominadas redes LLNs (Low-Power and Lossy Networks). Este estándar define las capas física y de acceso al medio. Las LLNs tienen requerimientos de transporte de datos mucho menores en comparación con las redes WLAN IEEE 802.11.

Contemplando los requerimientos de la creciente IoT y las restricciones de las LLNs, la IETF desarrolló el protocolo 6LoWPAN [3] en el estándar RFC-4944.

Este posibilita el uso del protocolo de Internet IPv6 en redes LLNs y define un conjunto de adaptaciones de la pila de protocolos estándar IPv6 entre la capa de enlace y red, nombrada como *Adaptation Layer*.

Adicionalmente, IETF desarrolló estándares para las capas superiores, como Constrained Application Protocol (CoAP)[4], con bibliotecas disponibles en numerosos lenguajes de programación. Es deseable el desarrollo de aplicaciones con estándares abiertos, pues asegura interoperabilidad entre diferentes sistemas.

El grupo de investigación GridTics está trabajando en el desarrollo de tecnologías IoT desde hace varios años. Se pueden mencionar los siguientes proyectos: "Livres: Análisis y evaluación de características relevantes de las WSN aplicadas al manejo y sensado en agricultura de precisión"(2010-2012), "SIPIA: Estudio de campo de red de sensores inalámbricos para adquisición de parámetros ambientales, de uso en investigaciones agronómicas y biológicas"(2012-2014) y "GW-CIAA-IoT: Gateway con CIAA para red inalámbrica de IoT"(2015-2016). https://www.overleaf.com/6881371211tgsnhrnjcxpk

Este último proyecto realizaba la interconexión entre una red de sensores y una red con soporte IPv6, está completamente desarrollado en los siguientes trabajos científicos [5][6]. El gateway, realizado con la placa CIAA [7], ejecuta el sistema operativo FreeOSEK [8]. Al no contar con soporte para protocolos IoT, el trabajo implicó la implementación de los protocolos IPv6 y SLIP6. Por otra parte, el almacenamiento y presentación de los datos obtenidos en los nodos fue realizada en un host con arquitectura x86 y sistema operativo GNU/Linux conectado a la red IPv6. La debilidad de este esquema es la pérdida de información en caso de no haber conectividad entre el gateway y el host, pues los datos adquiridos no pueden ser almacenados. Este problema recién se puso de manifiesto en el despliegue de una red IoT en producción, al dejar el host de almacenamiento y presentación en el laboratorio del GridTics, y el gateway en campo. Para solucionar este inconveniente se realizó el traslado de dicho host a campo, pero esto retrazó su despliegue, pues se debió hacer acondicionamientos físicos y energéticos para instalarlo en campo, sin contar que sería necesario tener un proveedor de Internet con soporte IPv6 en el lugar.

Otra alternativa analizada para solucionar este inconveniente fue instalar todas las aplicaciones del host en el gateway. FreeOSEK no soporta lenguajes de programación de alto nivel, y solo se pueden desarrollar aplicaciones en lenguaje C. Tampoco existen bibliotecas para manejo de periféricos ni protocolos IoT, como CoAP, para facilitar el desarrollo. Además la comunidad que da soporte a este sistema operativo no es muy activa. Debido a lo anterior, esta alternativa fue descartada.

Por lo expuesto, el presente desarrollo es una mejora del trabajo anterior, solucionando los inconvenientes mencionados. El objetivo principal de este proyecto es desplegar un sistema de IoT compuesto por una red de sensores inalámbricos de bajo consumo empleando protocolos abiertos y estándares, y concentrar en un solo dispositivo todo el software necesario para implementar exitosamente los servicios requeridos.

El resto del trabajo se desarrolla de la siguiente manera. La sección 2 detalla los componentes con los que cuenta el sistema, en el sección 3 se comparten los resultados obtenidos y finalmente las conclusiones a las que se arribó en el presente trabajo.

2. Componentes del Sistema

Se enumeran y describen a continuación todos los componentes utilizados para constituir el sistema completo de IoT.

2.1. Red de sensores Inalámbricos

Los nodos que componen estas redes suelen estar limitados por el consumo energético al estar usualmente alimentados por baterías, lo que permite desplazarse en el área a controlar. La distancia entre estos nodos puede llegar hasta 100 metros con visión directa entre antenas, así se abarcan grandes extensiones con un número reducido de dispositivos. Cada nodo debe comunicarse al menos con otro de los nodos de la red hasta llegar al nodo coordinador, haciendo múltiples saltos y formando una red tipo malla. Debido que un nodo puede encontrar diferentes rutas para alcanzar el coordinador, este tipo de redes posee redundancia, y si algún nodo falla o agota su energía, el resto puede encontrar una ruta alternativa para llegar al coordinador.

Para el diseño de la red fueron usados nodos sensores OpenMote CC2538, los cuales poseen sensores de temperatura, humedad y luz, entre otros. A su vez disponen de zócalos adaptadores para conexión USB y alimentación por baterías [9]. Los mismos ya estaban disponibles por haber sido usados anteriormente en otro proyecto.

Los nodos utilizan el estándar IEEE 802.15.4 el cual define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos para el ahorro máximo de energía. Se utilizó el estándar 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) en la subcapa de adaptación de red, lo que posibilitó el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4. Esto permite que los nodos de la red inalámbrica puedan comunicarse directamente y extremo a extremo con otros dispositivos IPv6.

El protocolo usado en capa de red fue IPv6, donde su gran capacidad de direccionamiento es particularmente útil en este tipo de redes dada la cantidad de nodos sensores que pueden ser utilizados. Por otra parte, el uso de este protocolo elimina la necesidad de servicios como DHCP y NAT.

En la actualidad existen desarrollos y soluciones que permiten el despliegue de redes de sensores basadas en los protocolos seleccionados. Entre las posibles soluciones, se seleccionó el sistema operativo Contiki OS [10] para ser ejecutado en cada nodo de la red. Sus principales ventajas comparativas son compatibilidad con 6LoWPAN, stack completo de TCP/IP y RPL.

La figura 1 muestra los nodos sensor y coordinador.

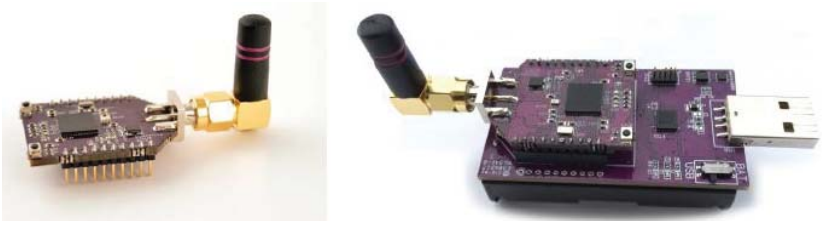

Figura 1. Nodo sensor (izquierda) y nodo coordinador (derecha).

2.2. Servidor de datos

Esta aplicación es ejecutada en cada uno de los nodos. El objetivo es proveer valores de temperatura ambiente, humedad e iluminación cada vez que un cliente le haga una petición.

Inicialmente se utilizó un stack completo de protocolo TCP/IP y soporte para HTTP-REST, pero los resultados obtenidos no fueron satisfactorios. Este tipo de redes tiene un alto porcentaje de probabilidad de pérdida. Por tanto, el uso de HTTP-REST sobre TCP/IP resulta desaconsejable.

Se sustituyó por una implementación de protocolo CoAP/UDP por su rapidez y eficiencia. Es un protocolo con arquitectura REST, heredado de HTTP. Utiliza las mismas primitivas de servicio (GET, POST, PUT, DELETE). A diferencia de HTTP, CoAP distingue mensajes confirmados y no confirmados. Los mensajes confirmados requieren un reconocimiento por parte del receptor, los no confirmados no. Fue seleccionada la primera opción para asegurar que los mensajes lleguen a destino a pesar de funcionar sobre un protocolo no orientado a la conexión como es UDP.

Adicionalmente, CoAP codifica los mensajes en formato binario en lugar del formato texto empleado en HTTP, lo que supone una reducción significativa de la longitud de las cabeceras y del propio mensaje. Esto permite un mejor aprovechamiento del ancho de banda disponible.

En capa de transporte, se utilizó UDP por su sencillez y reducido tamaño de cabecera. No fue necesaria la utilización de un protocolo orientado a conexión ya que no era necesario disponer control de flujo y confirmación de entrega o recepción. Cabe aclarar que Contiki provee bibliotecas para utilizar UDP y CoAP.

2.3. Nodo Coordinador o Router de Borde

El nodo coordinador es un nodo más de la red inalámbrica, pero posee la funcionalidad de mantener actualizadas las rutas para llegar a cada uno de los nodos sensores de la red. El nodo coordinador difunde el prefijo de la red IPv6, administra y registra los dispositivos de la WPAN mediante radiofrecuencia.

Los nodos OpenMote permiten programar uno de los nodos con adaptador USB como router de borde. El software consulta a los nodos sensores su dirección

MAC y a partir de un prefijo de red establecido, construye y difunde las diversas direcciones IPv6 globales. Tiene la capacidad de soportar la topología de red en malla, por lo que cada nodo está conectado con uno o más nodos de la red. Esta topología reduce el riesgo de fallos, y por ende el mantenimiento periódico. En caso que un nodo falle, los nodos adyacentes propagarán un cambio en la tabla de rutas, notificando a nodos contiguos del cambio en la red, y así sucesivamente, dando confiabilidad.

Se destaca que Contiki utiliza RPL (Routing Protocol for Low Power and Lossy Networks) [11], un protocolo de enrutamiento para redes inalámbricas de bajo consumo de energía y típicamente susceptibles a pérdidas de paquetes. Es un protocolo proactivo basado en vectores de distancia y opera sobre IEEE 802.15.4. Se basa en el modelo multisalto, en el que un nodo que transmite a la estación base, lo hace reenviando sus datos a uno de sus vecinos hasta que llega al destino.

La figura 2 muestra un ejemplo de una red IoT con RPL.

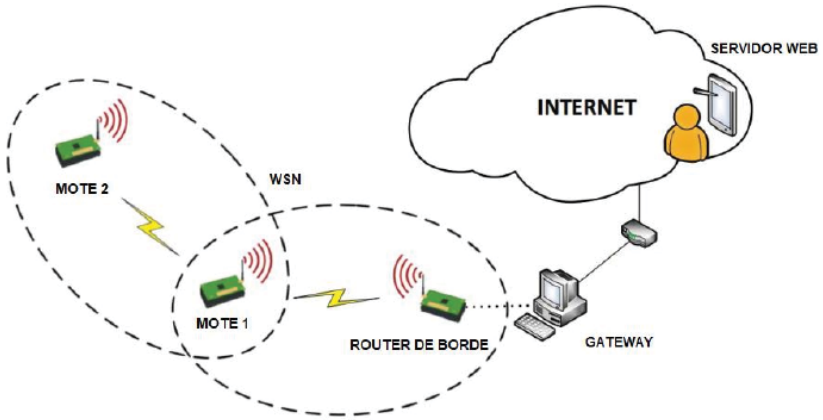

Figura 2. Ejemplo de red IoT con RPL.

Finalmente, Contiki soporta Neighboor Discovery, el cual es un protocolo que mediante la utilización de mensajes especiales ICMPv6 cada dispositivo conoce las direcciones IP del resto de los miembros de la red usando un mecanismo de autoconfiguración, sin necesidad de servicios como DHCP.

2.4. Gateway

Este dispositivo está conectado al coordinador utilizando la interfaz USB, y a Internet (IPv6) por medio de una interfaz Ethernet o WiFi. Su función es adaptar el protocolo IPv6 de Internet a la capa de enlace de la red sensores (802.15.4).

La implementación del gateway se realizó en una placa Rapsberry Pi [12], seleccionada de acuerdo a los objetivos de este proyecto. Raspberry Pi es una computadora de bajo costo y tamaño reducido, del orden de una tarjeta de crédito, desarrollada con el objetivo de estimular la enseñanza de la informática en las escuelas. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. En este proyecto se utilizó la Raspberry Pi 3 B+ que cuenta con las siguientes características: Processor Cortex-A53 (ARMv8) SoC de 64 bits a 1,4 GHz, 1 GB de RAM, LAN inalámbrica IEEE 802.11 de 2.4GHz y 5GHz, Bluetooth 4.2, Gigabit Ethernet, video HDMI , 4 puertos USB 2.0 y micro SD para cargar su sistema operativo y almacenar datos, entre otras cosas.

Esta placa no incluye sistema operativo alguno, se opto por instalar Raspbian-OS basado en Debian, por estar optimizado para el hardware de Raspberry Pi.

La figura 3 muestra una Raspberry Pi con un nodo actuando como router de borde.

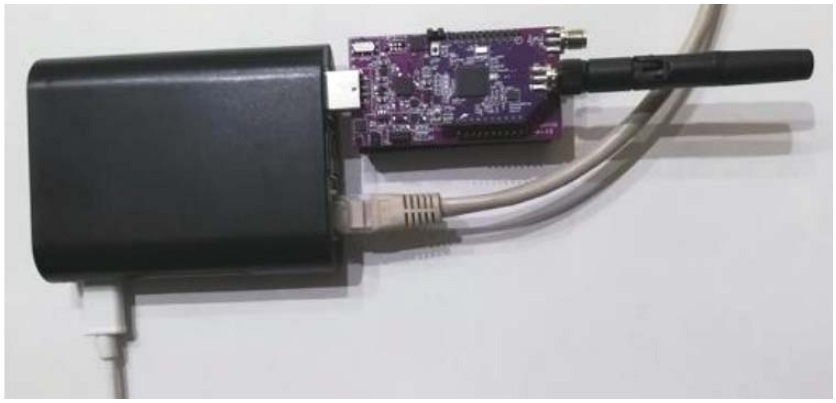

Figura 3. Raspberry Pi con nodo coordinador o router de borde.

La aplicación tunslip6 [14] se ejecuta en el gateway. Establece una conexión hacia el puerto serie (USB) y genera una interfaz virtual IPv6 que habilita la comunicación de paquetes desde y hacia la red de sensores. Además, permite administrar los parámetros de red IPv6 global que le serán otorgados al router de borde.

2.5. Cliente de datos

Este programa se ejecuta en el gateway y realiza periódicamente una petición de datos a cada nodo, con un periodo que puede ser ajustado por el usuario, como se detallará más adelante. El servidor de datos que se ejecuta en los nodos, al recibir esta solicitud, mide las variables requeridas y luego le responde con dicha información en un formato preestablecido (CoAP). Esta comunicación puede

ser punto a punto, en el caso de que el nodo esté en el rango de visión del coordinador, o multisaltos en caso que el nodo se comunique con el coordinador por medio de otros nodos.

El cliente se desarrolló usando el lenguaje de programación Python [13], cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Posee una licencia de código abierto compatible con la Licencia pública general de GNU. Este lenguaje es interpretado, multiparadigma y multiplataforma. Contiene una gran cantidad de bibliotecas, tipos de datos y funciones incorporadas en el propio lenguaje, que facilitan la realización de muchas tareas comunes. Las bibliotecas más importantes usadas en este proyecto fueron Coapthon, para implementar el protocolo CoAP, InfluxDB para manejo de bases de datos, y Request para trabajar HTTP/1.1 en Python.

Con la ayuda de estas bibliotecas, el programa cliente configura los nodos asociados a la red, realiza periódicamente peticiones CoAP a los nodos, solicitando los datos adquiridos por los sensores de los mismos, interpreta la trama de respuesta de los nodos, identifica los valores de los parámetros deseados y guarda dicha información en la base de datos influxDB.

El periodo de estas peticiones, es programable por parte del usuario a través de la plataforma de visualización.

2.6. Almacenamiento de datos

Cada vez que la aplicación cliente de datos obtiene información de los nodos sensores, se guarda en un base de datos no relacional. Esta base de datos también está instalada y ejecutandose en el gateway. Se seleccionó InfluxDB [15] debido a que es una base de datos basada en series de tiempo (time-series database), no relacional y de código abierto. Está optimizada para el almacenamiento rápido y de alta disponibilidad de datos de series temporales en campos como el monitoreo de operaciones, la métrica de aplicaciones, los datos de sensores de Internet de las cosas y el análisis en tiempo real. El formato de almacenamiento de datos en InfluxDB es JSON (JavaScript Object Notation), el cual es un formato de texto ligero para el intercambio de datos.

2.7. Servidor de visualización

El servidor permitir la visualización y control de los datos adquiridos por los nodos sensores, y la programación de los tiempos de adquisición de datos, alertas. Además, permite hacer comparaciones entre las variables medidas.

Grafana [16] es la plataforma de visualización seleccionada. Es un software para la visualización, análisis y supervisión de datos. Se trata de un servidor web de código abierto y multiplataforma. Permite crear paneles personalizados para la visualización de datos, configurar alertas y enviar notificaciones. Grafana soporta un variado tipo de bases de datos para la adquisición de información, incluyendo InfluxDB. El servidor de visualización también fue instalada en el gateway.

2.8. Cliente de visualización

Como cliente de visualización se utiliza un navegador o browser con soporte de IPv6. Por ello, la máquina en la que se ejecute debe poseer una dirección IPv6 global, para poder acceder al gateway.

3. Resultados

En la figura 4 se muestra esquemáticamente el diseño final de la red. Se detallan la pila de protocolos y aplicaciones utilizados en cada componente del sistema.

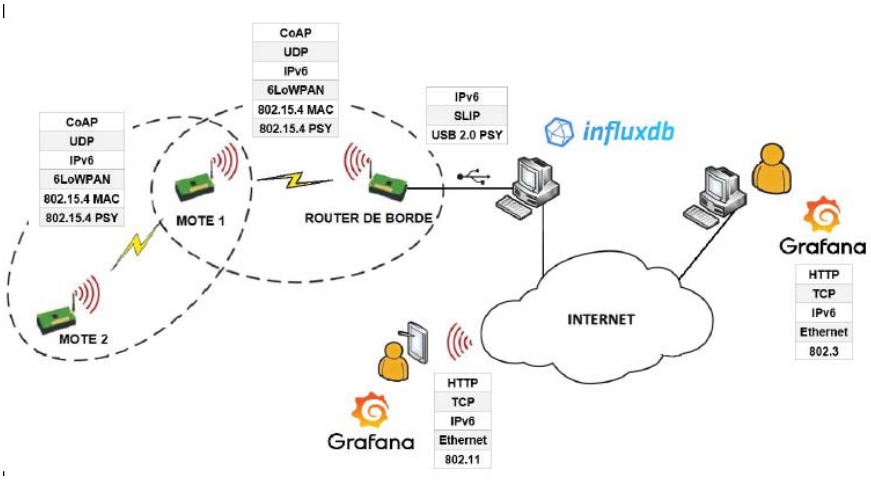

Figura 4. Sistema completo de la red IoT.

Se ensayó exitosamente una red compuesta por 4 nodos sensores OpenMote, un coordinador OpenMote y un gateway implementado con Rasperry Pi. Se obtuvieron distintos gráficos y se observaron las variables ambientales almacenadas.

A continuación, la figura 5 muestra algunas capturas de los paneles creados en Grafana para la visualización de las magnitudes sensadas por la red IoT.

La figura 6 muestra cómo se puede modificar el tiempo de refresco para la adquisición de datos modificando el valor en la solapa *refresing every*.

Se pudo verificar, tanto en un test bed, como en un despliege en campo, que las perdidas de conectividad no provocaron pérdidas de datos, pues los mismos se siguieron almacenando en la base de datos. Adicionalmente, se ensayó el correcto funcionamiento de todo el sistema, usándolo continuamente durante aproximadamente 1 mes en el testbed de GridTics.

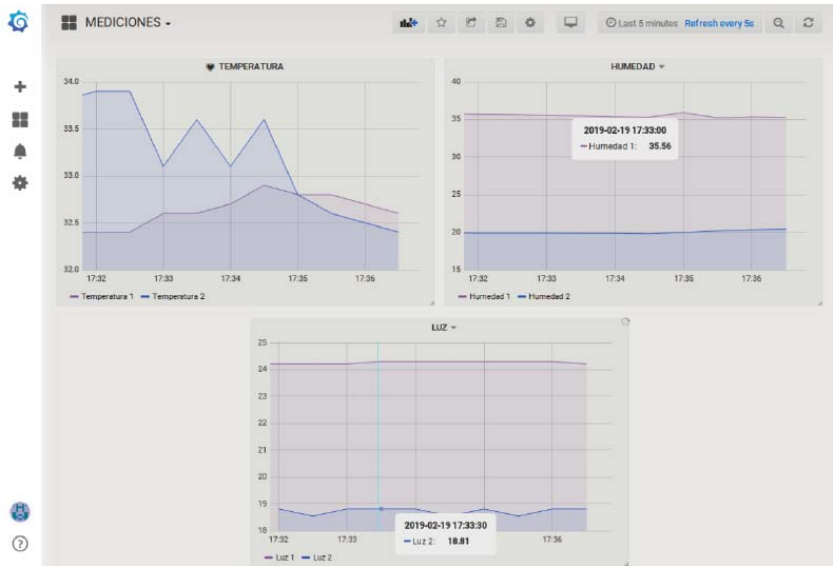

Figura 5. Paneles creados en Grafana para la visualización de diferentes variables ambientales.

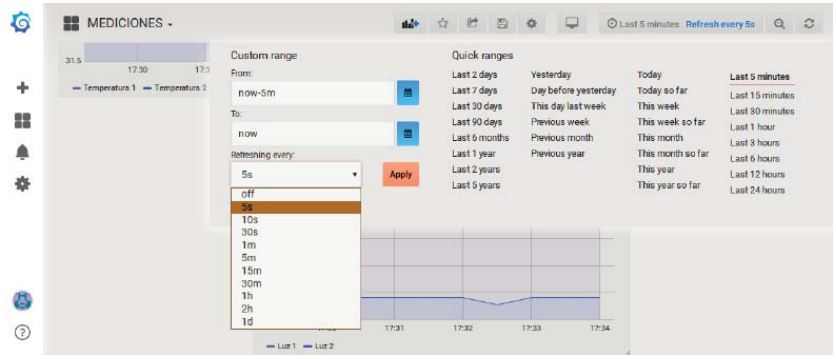

Figura 6. Panel en Grafana para ajustar el tiempo de refresco de las variables ambientales.

4. Conclusiones

A pesar del lento despliegue que está teniendo IPv6 en Internet, el número de dispositivos que demandan este servicio es cada vez mayor. Esto promueve que la transición hacia el nuevo protocolo de red sea ya una realidad inminente. Con la solución que provee este trabajo, los nodos de la red son dispositivos con direcciones IPv6 propias y cuentan a priori con conectividad extremo a extremo con cualquier sistema de la red Internet.

La selección de la plataforma Raspberry Pi, fue fundamental en el éxito del presente trabajo, pues concentra en un solo dispositivo diferentes componentes para el correcto funcionamiento y manejo de una red IoT. Los componentes integrados en ella son: el coordinador de red de sensores, un gateway IPv6, un cliente de datos, el almacenamiento de datos y un servidor de visualización.

Contiki fue escogido como sistema operativo a ejecutar en los nodos sensores. La ventaja respecto a otros software fue la posibilidad de usar un conjunto de herramientas que facilitan el despliegue de diferentes escenarios y también poder realizar simulaciones.

Configurada y establecida la red de sensores, con el propósito de ofrecer servicios sobre los nodos de la red desplegada, se estudiaron e implementaron distintas alternativas para acceder a recursos que ofrecen los nodos, así como un sistema de recolección y almacenamiento de datos. En términos de eficiencia, la solución REST basada en CoAP presenta mayor rendimiento frente a la tradicional empleando HTTP.

Los ensayos de confiabilidad de los datos frente a problemas de conectividad fueron exitosos en todos los casos. Adicionalmente se verificó que el despliegue de la red en pruebas de campo y laboratorio fue mas sencilla, debido a que una buena parte del software estaba integrado en el dispositivo gateway.

Referencias

1. Waher, P.: Learning Internet of Things, 1st Edition. Packt Publishing, UK (2015).
2. Molisch, A.F., Balakrishnan, K., Chong, C.C., Emami, S., Fort, A., Karedal, J., Kunisch, J., Schantz, H., Schuster, U. and Siwiak, K.,: IEEE 802.15. 4a channel model-final report. IEEE P802, 15(04), p.0662 (2004).
3. Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals (2007).
4. Shelby, Z., Hartke, K., Bormann, C.: The constrained application protocol (CoAP) (2014).
5. Taffernaberry, J.C., Mercado, G., Pecchia, M., Tobar, S., Verdejo, A., Sayago, J.: Puerta de Enlace para Internet de las Cosas usando Computadora Industrial Abierta. In: XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018), pp. 174-179, ISBN 978-987-3619-27-4. Corrientes, Argentina (2018).
6. Taffernaberry, J.C., Mercado, G.: GW- CIAA-IoT: Gateway con CIAA para red inalámbrica de IoT. In: WICC 2016, ISBN: 978-950-698-377-2. San Juan, Argentina (2016).
7. Proyecto CIAA. Disponible en http://www.proyecto-ciaa.com.ar. Mayo 2019.
8. Proyecto FreeOSEK. Disponible en http://opensek.sourceforge.net. Marzo 2019.
9. Vilajosana, X., Tuset, P., Watteyne, T. Pister, K.: OpenMote: Open-source prototyping platform for the industrial IoT. In: International Conference on Ad Hoc Networks (pp. 211-222). Springer, Cham (2015).
10. Contiki: The Open Source OS for the Internet of Things. Disponible en https://www.contiki-os.org. Mayo 2019.
11. Winter, T.: RPL: IPv6 routing protocol for low-power and lossy networks. IETF (2012).

12. Raspberry Pi. Disponible en https://www.raspberrypi.org. Mayo 2019.
13. Python. Disponible en https://www.python.org. Mayo 2019.
14. Romkey, J.L.: Nonstandard for transmission of IP datagrams over serial lines: SLIP. RFC 1055 (1988).
15. InfluxDB open-source time series database. Disponible en https://www.influxdata.com. Mayo 2019.
16. Grafana: The open platform for beautiful analytics and monitoring. Disponible en https://www.grafana.org. Mayo 2019.