

Desarrollo y evaluación de un gateway móvil IoT para redes 4G LTE

Developing and evaluation of an IoT mobile gateway for 4G LTE networks

Manuel Quiñones-Cuenca¹, Hever Patricio Pachar Bravo², Javier Martínez-Curipoma³, Luis Quiñones⁴, Rommel Torres⁵

Resumen

La tendencia actual es interconectar cualquier dispositivo para ser monitoreado y controlado de manera remota por medio de tecnologías propias del concepto de IoT. Uno de los problemas es conectar los dispositivos a una red mediante tecnologías con suficiente cobertura y capacidad, por lo que en el presente artículo se muestra el desarrollo de un *gateway* móvil para aplicaciones IoT bajo la plataforma Android, evaluado en redes móviles de 4G. El *gateway* móvil IoT envía los datos recolectados de sensores mediante la selección de los protocolos MQTT y HTTP, hacia plataformas IoT en Internet. La arquitectura se compone de sensores de temperatura, luminosidad y presión que se integran en un nodo sensor, cuya función es la obtención de los datos, procesamiento y luego, mediante Bluetooth, se envía a un *smartphone* Android, donde se ejecuta una aplicación que hace la función *gateway* móvil. Aquí se realiza el entramado de datos que son enviados mediante la red 4G LTE hacia las plataformas IoT de Ubidots y Watson IoT. Posteriormente se realizaron pruebas con varias métricas como latencia, demanda de corriente y costos de conexión. Se concluye que el *gateway* móvil IoT tiene un mejor rendimiento bajo el protocolo MQTT, en comparación con HTTP.

Palabras clave

Internet de las Cosas, MQTT, HTTP, Ubidots, Watson IoT, WSN, LTE

Abstract

The current trend is to connect everything to network to be monitored and controlled remotely through own concept IoT technologies. One of the problems is to connect the devices to a network, using technologies with sufficient coverage and capacity, so this article shows the development of a mobile gateway for IoT applications under the Android platform evaluated in 4G mobile networks. The IoT mobile gateway sends the data collected from sensors by selecting the MQTT and HTTP protocols, to IoT platforms on the Internet. The architecture is composed of temperature, luminosity and pressure sensors that are integrated into a sensor node which function is for data collection, processing and then via Bluetooth is sent to an Android smartphone where an application that performs the mobile gateway function is running, here data network are sent through the 4G LTE network to the IoT platforms of Ubidots and Watson IoT. Subsequently, tests were carried out with various metrics such as latency, energy consumption and connection costs. It concludes that IoT mobile gateway has better performance under MQTT protocol compared to HTTP.

Keywords

Internet of things, MQTT, HTTP, Ubidots, Watson IoT, WSN, LTE

1. Introducción

Actualmente, IoT (Internet of Things) o Internet de las Cosas, es un concepto que tiene por objetivo conectar todo o casi todo a Internet, para ser controlados, monitoreados o para tele-

1 Universidad Técnica Particular de Loja (UTPL), Ecuador. [mfquinones@utpl.edu.ec, <https://orcid.org/0000-0002-2932-1524>]
2 Universidad Técnica Particular de Loja (UTPL), Ecuador. [hppachar@utpl.edu.ec, <https://orcid.org/0000-0001-7124-2232>]
3 Universidad Técnica Particular de Loja (UTPL), Ecuador. [jfmartinez1@utpl.edu.ec, <https://orcid.org/0000-0001-6981-3323>]
4 Universidad Técnica Particular de Loja (UTPL), Ecuador. [lquinones@utpl.edu.ec, <https://orcid.org/0000-0001-5894-9882>]
5 Universidad Técnica Particular de Loja (UTPL), Ecuador. [rovitor@utpl.edu.ec, <https://orcid.org/0000-0003-2313-0118>]

metría (Cabrera-Goyes & Ordoñez-Camacho, 2018; Rodríguez et al., 2019). IoT es el resultado de años de evolución de Internet, donde el número de dispositivos conectados a Internet, según Zimmermann & Empspak (2017), eran 20 000 dispositivos en el año 1987, y en la actualidad existen más dispositivos conectados a Internet que personas en el planeta. La interacción entre dispositivos aumenta cada día. De acuerdo a Cisco (2016), en un futuro serán 500 mil millones de dispositivos conectados para el 2030.

Para la transmisión de datos en IoT, se puede aplicar una diversidad de protocolos y tecnologías de comunicación alámbricas e inalámbricas. Algunas de las tecnologías inalámbricas, como Zigbee, Bluetooth, Digimesh e incluso Wi-Fi, requieren que los dispositivos o nodos se emparejen con un controlador o un *gateway*, con IP habilitado para obtener acceso indirecto a Internet, mediante otras tecnologías de comunicación como las redes móviles o de servicio móvil avanzado (SMA) de 2G, 3G, y 4G. De estas, actualmente las redes móviles basadas en 4G usando LTE son las que, a nivel global, regional y en el Ecuador, más suscripciones tienen (Arcotel, 2017). Por esto la importancia de estudiar más a fondo el rendimiento de la red 4G en aplicaciones IoT.

Según Open Signal (2018), la disponibilidad de redes 4G en Latinoamérica varía entre el 80 % como máximo, en Uruguay; 43 % para el mínimo en El Salvador y 45% para Ecuador. La velocidad de acceso máxima la tiene Ecuador con un promedio de velocidad 23.3 Mbps, y en todos los países analizados se sobrepasa la barrera de los 10 Mbps. Por lo tanto, estas redes brindan un acceso adecuado para conectar dispositivos IoT, ya que actualmente están en expansión de cobertura y altas tasas de velocidad, a diferencia de las tecnologías inalámbricas para IoT basadas en Wi-Fi y Bluetooth, que tienen un menor alcance y cobertura (Majdi, 2013).

En el año 2014 la tecnología 4G LTE fue introducida a Ecuador por la estatal CNT EP (Arcotel, 2017), a diferencia de Brasil, Colombia y Uruguay, que han venido trabajando desde el año 2012. En estos países algunos trabajos relacionados plantean el uso de la red 4G LTE como un *gateway* para escenarios de IoT. Asimismo, Díaz-Zayas et al. (2016), proponen un análisis a la estandarización planteada por el 3GPP de añadir el soporte aplicaciones IoT bajo LTE. En este trabajo se explica la evolución de LTE para soportar IoT, lo que ha llevado a un aumento de la vida útil de la batería de los dispositivos IoT en el orden de años, reducida complejidad, mayor cobertura y alta densidad de nodos. Con respecto a los protocolos planteados HTTP y MQTT, en Sthepen (2012) se establece que HTTP consume más energía por hora, que MQTT, y los mensajes enviados usando MQTT son en promedio diez veces más que con HTTP; concluye que MQTT es más liviano al reducir el consumo de energía y más rápido. En Yokotani & Sasaki (2016) se establece con respecto al ancho de banda y consumo de recursos, que MQTT tiene un rendimiento superior que HTTP.

En este trabajo se plantea el diseño e implementación de un sistema donde se integra una aplicación en la plataforma Android, que permite establecer una puerta de enlace móvil (*gateway* móvil), entre una red de sensores inalámbrica (WSN) basada en Bluetooth y una red móvil 4G, mediante la selección de un protocolo de capa de aplicación (HTTP o MQTT) para la transmisión de los datos. La importancia del desarrollo del sistema radica en que compara la latencia, consumo de energía y costes de conexión, entre los protocolos HTTP y MQTT, cuando interactúan con plataformas de IoT. El sistema además ofrece un modelo escalable y funcional, con la capacidad de agregar nuevos nodos de sensores u otras tecnologías.

Este documento está estructurado de la siguiente manera: en la sección 2, Metodología, se plantea la arquitectura del sistema, requerimientos de hardware, protocolos de IoT y plataformas de IoT utilizadas. En la sección 3, se muestra la implementación de un prototipo de

una WSN y un *gateway* móvil, con la transmisión de datos a una plataforma IoT, usando HTTP y MQTT, y el análisis de los resultados obtenidos. Finalmente, en la sección 4 se presenta las conclusiones y recomendaciones.

2.2. Metodología

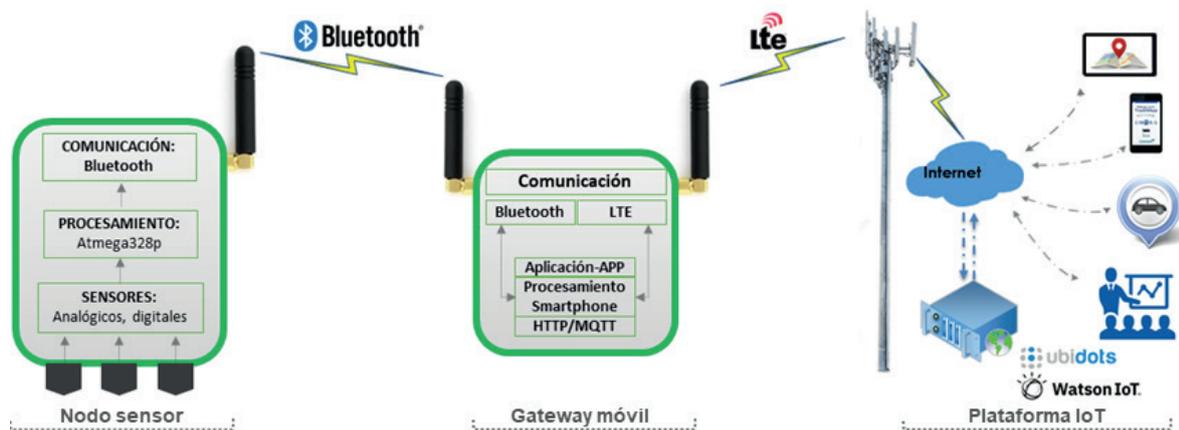
A través de un diseño experimental y un enfoque cuantitativo, se procede a realizar las siguientes fases de desarrollo e implementación del sistema:

1. Revisión del estado del arte de tecnologías disponibles para las WSN e Internet de las cosas IoT.
2. Diseño e implementación de la arquitectura de la red WSN que permite obtener los datos a ser transmitidos por la red 4G.
3. Desarrollo de la aplicación mediante Android Studio, y se realizan pruebas para validar el rendimiento de los dos tipos de protocolos a implementar (HTTP y MQTT) para el envío de los datos a plataformas IoT.
4. Análisis de los resultados de rendimiento con las métricas de consumo de energía, la latencia y costos de conexión, para establecer el protocolo que tiene mejores resultados sobre redes 4G.

2.1. Arquitectura del sistema

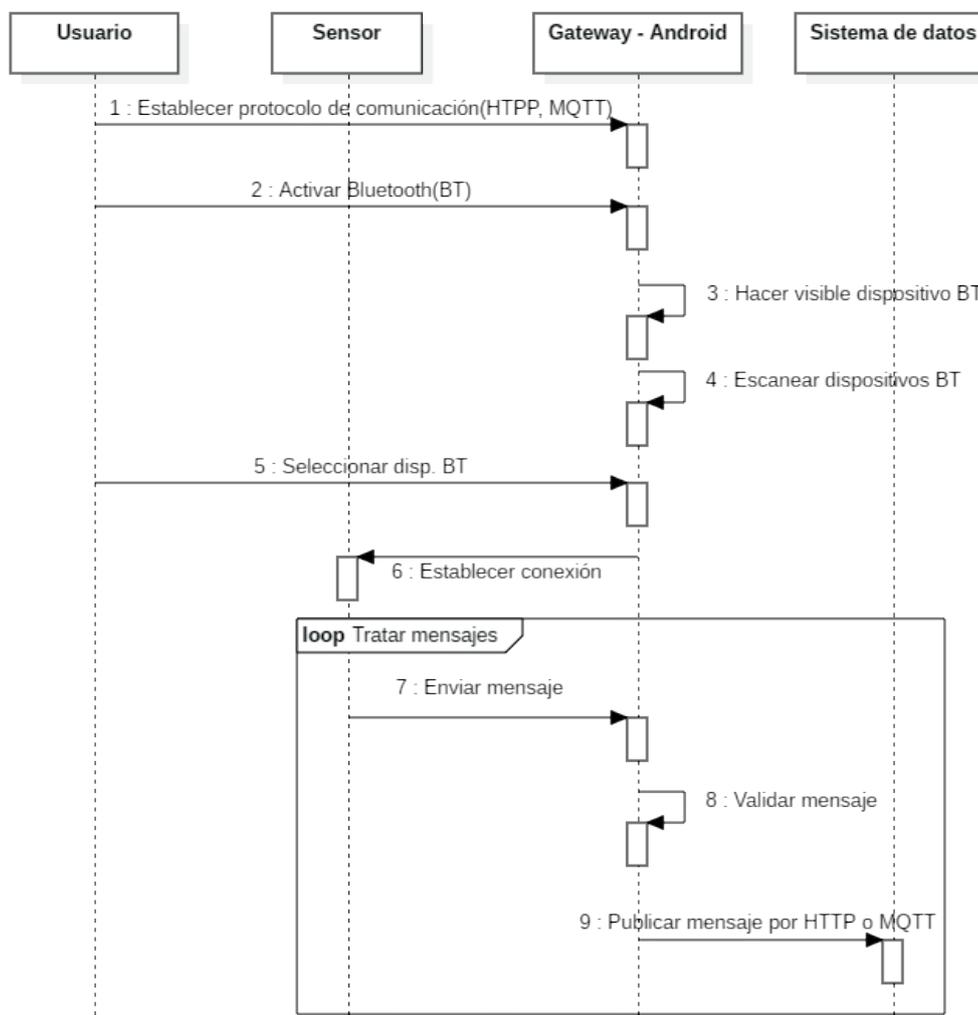
El sistema está integrado de elementos de hardware y software. En la figura 1 se detallan los tres componentes del sistema: Nodo sensor, *gateway* móvil y las plataformas de IoT.

Figura 1. Arquitectura del sistema



El sistema incorpora una red de sensores para obtener los datos de prueba de variables como temperatura, humedad y luminosidad; estos datos se envían por medio de Bluetooth a un *smartphone* que realiza las funciones de un *gateway* móvil. Para la implementación de este se desarrolla una aplicación en Android, donde se procesa las tramas recibidas desde la WSN y se visualiza los datos de cada variable. Luego esta aplicación envía los datos a dos plataformas de IoT mediante la selección de uno de los protocolos HTTP o MQTT. En la figura 2, mediante un diagrama de secuencias, se detalla la interacción de los distintos componentes del sistema.

Figura 2. Diagrama de secuencias: interacción entre los componentes del sistema



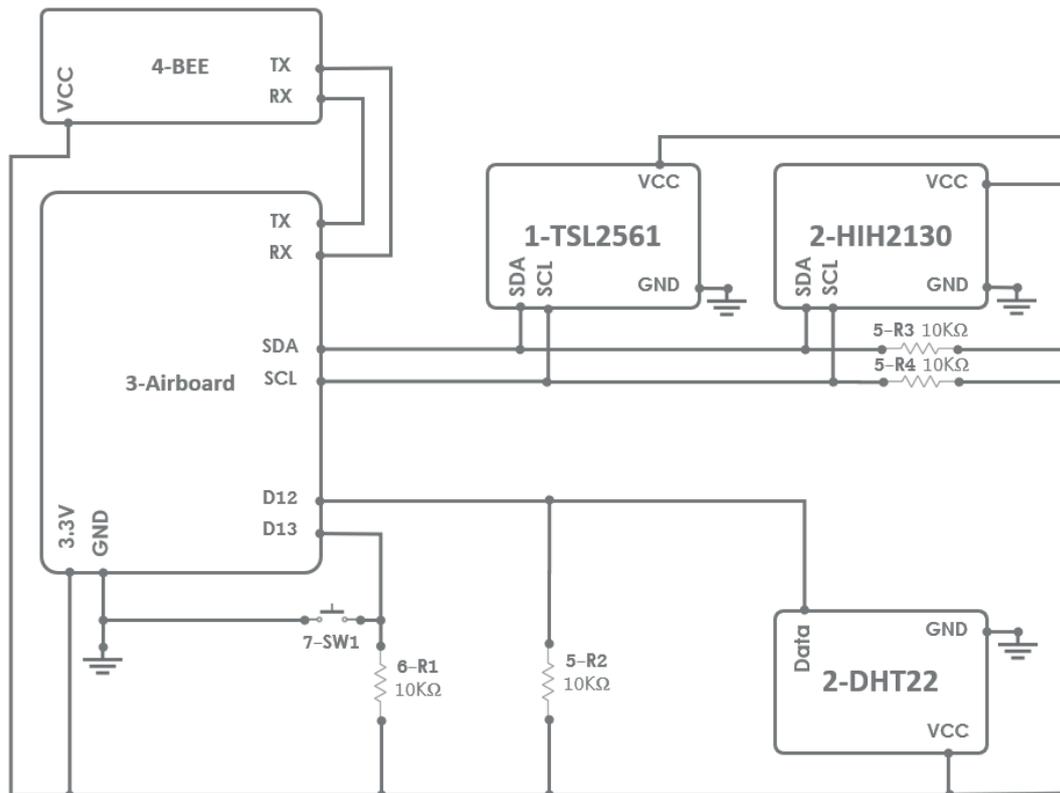
2.1.1. HARDWARE

En esta subsección se presenta los requerimientos de hardware para operar la aplicación del *gateway* móvil en el *smartphone*, y los componentes electrónicos utilizados para desarrollar el hardware del sistema embebido (nodo sensor). Para la operación del aplicativo móvil del *gateway* es necesario un *smartphone* que soporte el sistema operativo Android en la versión 4.4.4 o superior; con al menos las siguientes características 1 GB mínimo de RAM, conexión a datos móviles mediante LTE y procesador Quad-Core de 1.3 GHz. El nodo sensor está compuesto de los siguientes componentes electrónicos: 1) sensor de luminosidad (TSL2561), 2) sensores de temperatura y humedad relativa (DHT22 y HIH-6130), 3) plataforma de procesamiento para el nodo sensor (Airboard); y 4) módulo de comunicación Bluetooth (SainSmart Bee).

En la figura 3 se detalla el esquema de la arquitectura del hardware del nodo sensor, donde: el Airboard incorpora un microcontrolador que se usa para procesar la información obtenida por los sensores de luminosidad, temperatura y humedad relativa, y enviar la información por medio de tramas al módulo Bluetooth. El módulo Bluetooth es responsable de enviar la infor-

mación al *gateway* móvil; y el pulsador (SW1) permite iniciar la transmisión de las tramas. Los sensores integrados en el sistema son digitales y usan protocolos de comunicación como I2C y Single-bus Signal; y para la adecuada transmisión de los datos de los sensores al Airboard se requiere de resistencias de 10 k Ω en modo *pull-up* en la línea de transmisión de datos de acuerdo con cada protocolo de comunicación.

Figura 3. Esquema del nodo sensor



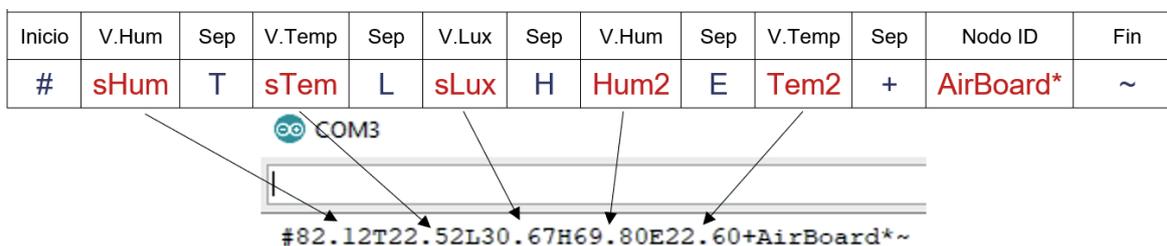
2.2. Software

En esta sección se detalla el funcionamiento y las herramientas utilizadas para el desarrollo del software para el aplicativo móvil y del nodo sensor.

2.2.1. ALGORITMO DEL NODO SENSOR

El algoritmo para el nodo sensor tiene las funciones de procesar y gestionar la información obtenida mediante los sensores, y estructurar las tramas, para su posterior envío al *gateway* móvil por medio del módulo Bluetooth. El algoritmo se desarrolla en el lenguaje de programación C para Arduino o mediante el entorno de desarrollo integrado (IDE) de Arduino. Este permite procesar y armar las tramas con los datos obtenidos de los sensores y enviarlas cada 30 segundos por el módulo Bluetooth hacia el *gateway* móvil. Las tramas son de tipo *string* y tienen el formato que se detalla en la figura 4.

Figura 4. Formato de trama enviada desde el nodo sensor al *gateway* móvil

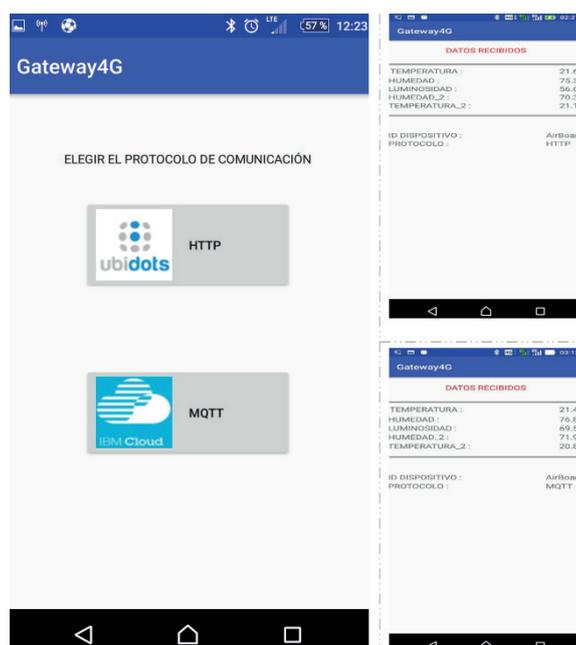


VHum: variable de humedad
VTemp: variable de temperatura
VLux: variable de luminosidad
Sep: separador

2.2.2. APLICACIÓN GATEWAY MÓVIL

La aplicación de la plataforma móvil es responsable de presentar la información de los sensores, permitir la selección del protocolo de IoT (HTTP o MQTT), gestionar la comunicación mediante Bluetooth con el nodo sensor, procesar las tramas recibidas desde el sistema embebido y transmitir la información mediante una red 4G LTE a la plataforma de IoT en base al protocolo seleccionado. Para el desarrollo de la aplicación móvil se utiliza el lenguaje de programación Java en el entorno de programación IDE de Android Studio. En la figura 5 se presenta la pantalla principal que permite seleccionar el protocolo de IoT. Para enviar la información a cada plataforma se usaron librerías de Ubidots para Android Studio (Ubidots, n.d.) y para la plataforma Watson IoT la librería de Eclipse PAHO MQTT v3 (Paho, n.d.).

Figura 5. Pantallas de la aplicación del *gateway* móvil para la selección protocolo de IoT



1.3. Protocolos de IoT usados

Actualmente, existe una variedad de protocolos de aplicación para IoT como MQTT (Message Queue Telemetry Transport), AMQP (Advanced Message Queuing Protocol), CoAP (Constrained Application Protocol) y HTTP (Hipertext Transfer Protocol) (Naik, 2017). En el sistema se evalúan los protocolos HTTP y MQTT; asimismo, se usan plataformas de IoT que empleen estos protocolos y presenten facilidades como acceso libre o temporal para el despliegue de las pruebas. A partir de estas características se determina las siguientes plataformas: Ubidots por medio de HTTP (Ubidots, 2019) y Watson IoT con MQTT (IBM, 2019).

1.3.1. HTTP

HTTP usa en el intercambio de páginas web y servidores mediante mensajes de petición/respuesta. También se aplica para otros fines, como en este caso para aplicaciones IoT. Usa mensajes compuestos de cabeceras que intercambian información entre cliente y servidor, estos mensajes de petición son: *get*, *put*, *post*, *delete* y *trace* (Kurose & Ross, 2017).

1.3.2. MQTT

MQTT es un protocolo para el intercambio de mensajes de suscripción y publicación, optimizado para aplicaciones que requieren el intercambio de mensajes de pequeña longitud. MQTT se adapta muy bien a los requerimientos de IoT, dado que los recursos de procesamiento, memoria y ancho de banda son limitados. Usa TCP como protocolo de transporte. Es utilizado en sensores con enlaces satelitales, conexiones telefónicas ocasionales para servicios de salud y en domótica con pequeños dispositivos. Este protocolo es óptimo para trabajar con aplicaciones móviles debido a sus prestaciones de bajo consumo de energía y transmisión de datos liviana (Hillar, 2017)

La arquitectura de MQTT permite tener una gran flexibilidad debido a su modelo de publicar y suscribir, donde el cliente es el suscriptor o el publicador. Estas actividades se realizan de manera asíncrona, las mismas que son dirigidas mediante mensajes a un nodo central llamado *Broker*, que hace de intermediario entre los publicadores y los suscriptores (Manandhar, 2017). Según lo indica IBM (2010), los mensajes que se intercambian en este protocolo son *connect*, *publish*, *subscribe*, *unsubscribe* y *disconnect*. En este sistema el *gateway* móvil hace la función de publicador y la plataforma Watson IoT de cliente.

3. Resultados y discusión

Esta sección presenta los resultados que se obtuvieron para determinar el rendimiento de los protocolos HTTP y MQTT implementados en el *gateway* móvil mediante la realización de los experimentos de demanda de corriente, latencia, y consumo de datos móviles. Por cada protocolo se realiza pruebas con una hora de duración para 5, 10, 15 y 20 solicitudes, donde cada solicitud se publica una variable. En la figura 6 se muestra la integración e implementación de los distintos componentes de hardware y software para el despliegue del sistema, descritos en la sección 2.1: el nodo sensor, el dispositivo móvil que hace la función de *gateway* móvil, y en el computador los datos que son registrados en una de las plataformas de IoT. En las siguientes subsecciones se realiza el análisis de los resultados.

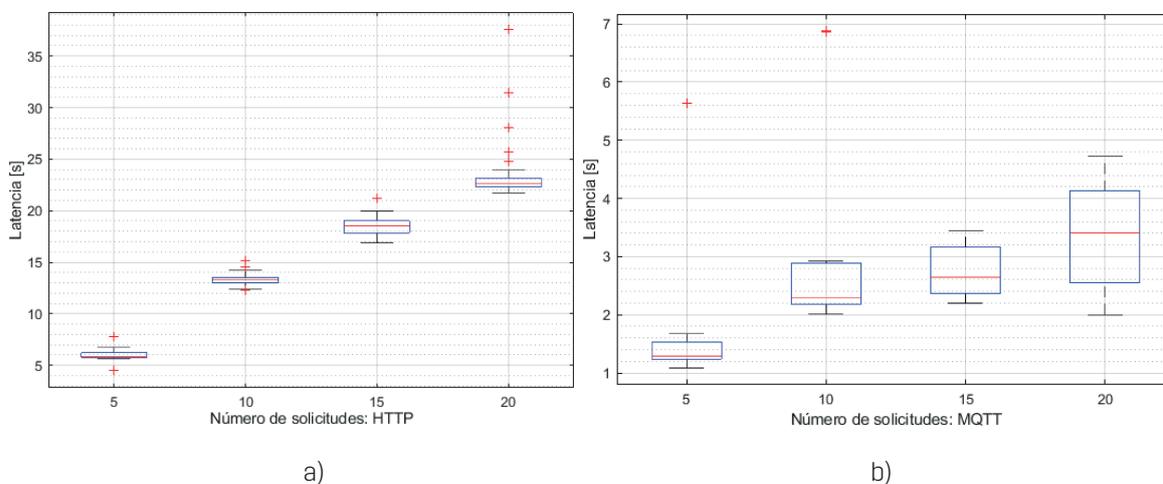
Figura 6. Implementación del sistema: nodo sensor, *gateway* móvil, y visualización del registro de datos en la plataforma Ubidots



3.1. Latencia

En las pruebas se usa un escenario para analizar el tiempo necesario para enviar 5, 10, 15 y 20 solicitudes cada 30 s, obteniendo 30 valores por cada prueba y por cada protocolo IoT, teniendo un conjunto 240 datos. Se entiende por latencia al tiempo transcurrido entre la primera solicitud y la recepción del último acuse de recibo de la última solicitud, es decir se ha despachado por completo estas transacciones. Para la captura de los paquetes se utilizó un *sniffer*, que genera un archivo tipo *pcap*, que contiene los datos del tráfico capturado y es analizado mediante Wireshark. Para el protocolo HTTP se usa el tiempo de transmisión desde la solicitud POST con el token de una cuenta en Ubidots hasta la última respuesta del servidor. Estos tiempos se pueden observar en el diagrama de caja de la figura 7.

Figura 7. Diagrama de cajas del tiempo de solicitud: a) HTTP y b) MQTT



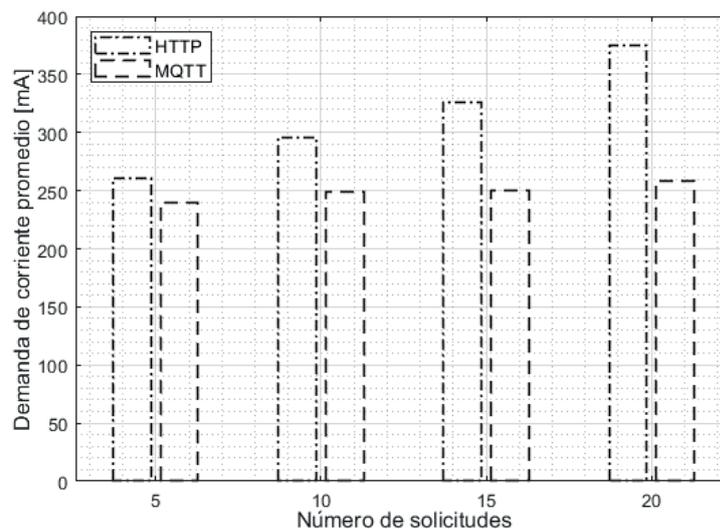
Se realiza una prueba Anova para determinar si la media de la latencia varía con respecto al número de solicitudes, con un nivel de significancia de 0.05, es decir un 95 % de confiabilidad. La hipótesis nula H_0 plantea que las medias de la latencia son significativamente distintas, con distinto número de solicitudes, mientras que la hipótesis no nula H_1 se afirma que, son significativamente iguales, esto se realiza con las muestras de los dos protocolos. Con los datos recolectados con el protocolo HTTP, usando la prueba de Fisher se obtuvo $F = 358.7592$ y un valor crítico $Z_c = 3.138$. Dado de $F > Z_c$ se rechaza la hipótesis H_0 , en consecuencia, se acepta la hipótesis nula H_1 , que establece que las medias de las distintas mediciones son significativamente distintas, con el protocolo HTTP.

Usando los datos para el protocolo MQTT, se obtiene $F = 67.1582$, y un valor crítico $Z_c = 3.162$, dado de $F > Z_c$, se concluye que las medias de las observaciones son significativamente distintas, pero en menor medida que en el protocolo HTTP. Finalmente, en HTTP se tiene la mayor cantidad de datos fuera de caja, según se muestra en la figura 7a, al crecer el número de solicitudes que permite prever una cierta inestabilidad en la red 4G, mientras que con MQTT se da en menor cantidad.

3.2. Demanda de corriente

Para analizar el consumo de corriente se ha usado la metodología planteada por Santos et al. (2016), donde se analiza el consumo de un *gateway* IoT en diferentes escenarios de consumo. Para obtener la información de la demanda de corriente se usa la aplicación Trepn Profiler (Qualcomm Technologies, 2019), que permite obtener datos del consumo eléctrico de la batería, carga de la CPU, frecuencia de la CPU, estado de la red móvil, entre otros datos. Las pruebas se realizaron en un *smartphone* Sony Xperia Z3 con un procesador Qualcomm MSM8974AC Snapdragon 801 con una conexión 4G LTE y una batería de Li-Ion de 3100 mAh. A partir de los datos obtenidos se determina que el tiempo de duración de la batería con MQTT oscila entre 12 a 13 horas, para 20 y 5 solicitudes respectivamente, mientras con HTTP el tiempo es de 8 (20 solicitudes) a 12 horas (5 solicitudes). El consumo de corriente promedio en las distintas observaciones y protocolos se establece en la Figura 8.

Figura 8. Media de la demanda de corriente en la batería en mA

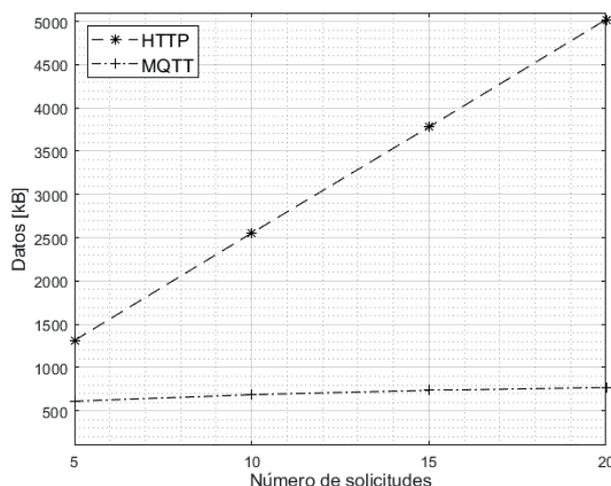


3.3. Consumo de datos móviles

Para determinar el consumo de datos móviles es necesario adquirir datos del tráfico que genera la aplicación; para este propósito se usa la herramienta App Tune-up Kit de Qualcomm (Qualcomm Technologies, n.d.), para analizar el rendimiento de los protocolos IoT mediante la variación del número de solicitudes enviadas cada 30 segundos, durante 60 minutos. Para el análisis del consumo de datos móviles se fija como referencia una tarifa de \$ 0.10/MB (Arcotel, 2018). De los datos recolectados se establece que el protocolo HTTP genera un tráfico con una desviación estándar de 1594.1 kB y un promedio de 3166.5 kB, y se establece una relación lineal, entre el tráfico y número de solicitudes.

En cambio, con el protocolo MQTT se genera tráfico con una desviación estándar de 69.3 kB y un promedio de 701.2 kB. A partir de este resultado se determina que el tráfico no varía al incrementar el número de solicitudes. De lo anterior se establece que HTTP genera mayor tráfico e implica un mayor costo en comparación con MQTT, en un 452 % en promedio. Finalmente, sobre la base del precio referencial de datos móviles, se determina que el costo del servicio para la prueba de 20 solicitudes con MQTT y HTTP está entre 0.08 y 0.50 dólares respectivamente. La cantidad de tráfico generado en las distintas observaciones se puede apreciar en la figura 9.

Figura 9. Tráfico generado en las distintas observaciones



4. Conclusiones y recomendaciones

Se diseñó e implementó un sistema que obtiene los datos de una WSN por medio de un *smartphone*, donde se ejecuta una aplicación que hace la función de un *gateway* móvil que permite procesar, determinar un protocolo de IoT (HTTP o MQTT) y enviar los datos a plataformas IoT mediante una red de servicio móvil avanzado 4G LTE. De los resultados de las pruebas de rendimiento, el protocolo MQTT presenta las mejores prestaciones en latencia, consumo de energía y datos móviles, al incrementar el número de solicitudes; esto implica una mayor disponibilidad de batería, menor tiempo y costo en la transmisión de datos. Además, el sistema permite flexibilidad para incrementar funcionalidades mediante la integración de otras tecnologías de comunicación y de protocolos de IoT para integrar más nodos, expandir la cobertura de la red y áreas de aplicación como mediciones inteligentes para servicios de agua y electricidad.

Finalmente, se recomienda analizar la integración de la función al *gateway* móvil, la capacidad de analizar los datos por medio de inteligencia artificial con el objeto de reducir el tiempo de respuesta, al no disponer de sistemas centralizados como las actuales plataformas de IoT.

Referencias

- Arcotel. (2017). *4G es la cuarta generación de tecnología de telefonía móvil*. Retrieved on March, 2020, from <https://bit.ly/3bqllMn>
- Arcotel. (2018). *Arcotel Informa No19*. Retrieved on March, 2020, from <https://bit.ly/3h8RFWx>
- Cabrera-Goyes, E., & Ordóñez-Camacho, D. (2018). Posicionamiento en espacios interiores con Android, Bluetooth y RSSI. *Enfoque UTE*, 9(1), 118–126. <https://doi.org/10.29019/enfoqueute.v9n1.238>
- Cisco. (2016). *Internet of Things*. CISCO-LABS. <https://bit.ly/3buPGdW>
- Díaz-Zayas, A.; García-Pérez, C. A.; Recio-Pérez, M. y Merino, P. (2016). 3GPP Standards to Deliver LTE Connectivity for IoT. *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 283–288. doi: 10.1109/IoTDI.2015.26
- Hillar, G. C. (2017). *MQTT Essentials: A Lightweight IoT Protocol: The Preferred IoT Publish-Subscribe Lightweight Messaging Protocol*. Packt Publishing.
- IBM. (2019). *IBM Watson Internet of Things*. Retrieved on March, 2020, from <https://ibm.co/2QZGfta>
- IBM, I. B. M. C. (2010). *MQTT V3.1 Protocol Specification*. Retrieved on March, 2020, from <https://ibm.co/3gUmo9y>
- Kurose, J. y Ross, K. (2017). *Redes de computadoras : Enfoque descendente* (7ª. ed.). Pearson.
- Majdi, M. (2013). *A comparative Overview of Modern Communication Systems and Standards*. KTH, School of Electrical Engineering (EES). <https://bit.ly/3gX4VgN>
- Manandhar, S. (2017). MQTT based Communication in IoT. *TREPO* <https://trepo.tuni.fi/handle/123456789/25376>
- Naik, N. (2017). Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. doi: 10.1109/Sys-Eng.2017.8088251
- Fitchard, K. (2018). As 4G reach in Latin America improves, so does the consumer experience. *Opensignal*. <https://bit.ly/2QXfxla>
- Paho, E. (2019). *Eclipse Paho Android Service*. Recuperado el 25 de septiembre de 2019 de <https://bit.ly/3h3Vdci>
- Qualcomm Technologies, I. (s. f.). *App Tune-up Kit*. Recuperado el 25 de septiembre de 2019 de <https://bit.ly/3i1ssOS>
- Qualcomm Technologies, I. (2019). *Treppn Power Profiler*.
- Rodríguez, D. M.; Bayona, E. y Rosado, A. A. (2019). Summary of the Internet of Things and its Application in Agro-industrial Production. *Journal of Physics: Conference Series 1409*, 12018. doi: 10.1088/1742-6596/1409/1/012018
- Santos, J.; Rodrigues, J. J. P. C.; Silva, B. M. C. et al. (2016). An IoT-based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments. *Journal of Network and Computer Applications 71*, 194–204. doi: 10.1016/j.jnca.2016.03.014
- Sthepen, N. (2012). *Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android*. Scoop.it. <http://sco.it/7IR1rV>
- Ubidots. (s. f.). *Android SDK*. Recuperado el 5 de agosto de 2019 de <https://bit.ly/3gYosNS>
- Ubidots. (2019). *Ubidots : Data Drives Decisions*. Retrieved on August, 2019 from <https://bit.ly/3i1mZHT>
- Yokotani, T. y Sasaki, Y. (2016). Comparison with HTTP and MQTT on Required Network Resources for IoT. *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 1–6. doi: 10.1109/ICCEREC.2016.7814989
- Zimmermann, K. A. y Empspak, J. (2017). Internet History Timeline: Arpanet to the World Wide Web. *LiveScience*. <https://bit.ly/3i4e9cj>