



Implementación de un Cliente MQTT como Pasarela de una Red de Sensores Inalámbricos (WSN)¹

Milton Eduardo Sosa²; Eduardo Omar Sosa³; Germán Andrés Xander⁴

¹Trabajo de Investigación

²investigador, mesosa@fceqyn.unam.edu.ar

³investigador, es@fceqyn.unam.edu.ar

⁴investigador, gaxander@fio.unam.edu.ar

Resumen

En el presente trabajo, se presenta una manera de integración de una red de sensores inalámbricos (WSN) implementada en la Universidad Nacional de Misiones, y un agente de mensajes (bróker), el que como una aplicación de Internet de las Cosas (IoT) realiza las actividades de un nodo concentrador/pasarela. El protocolo de conectividad es extremadamente liviano, basado en TCP/IP bajo método publicidad/ suscripción, que se organizan sobre la base de la noción de “tema”: MQTT (Message Queuing Telemetry Transport) el cual permite a los dispositivos establecer una conexión utilizando poca energía, y adicionando los mensajes con poco overhead de datos. Considerando que el hardware de los nodos de la red actual no posee capacidad de manejar en forma directa la pila TCP/IP, surge como propuesta utilizar el mismo nodo concentrador como agente intermediario de esta tarea. La solución propuesta consistió en un script de lenguaje Python corriendo en una computadora de bajo costo Raspberry Pi, que ha resultado un caso de éxito.

Palabras Clave: *Redes de Sensores Inalambricos; WSN, MQTT, Internet de las cosas*

Introducción

Internet de las Cosas (IoT) considera un ambiente de objetos unívocamente identificables, en el cual pueden coexistir diversos sensores que captan información del entorno volviéndola útil para el usuario. Esta información se presenta a quienes la solicitan a través de aplicaciones en dispositivos móviles de uso cotidiano, permitiendo tomar decisiones sobre acciones a ejecutar. Extrapolando el escenario propuesto a un nivel global, resulta en miles de millones de sensores cableados e inalámbricos formando una compleja red que genera un extraordinario tráfico de datos.

Impulsado por la conectividad ubicua y la disponibilidad de billones de direcciones IPv6 [1], el número de dispositivos conectados globalmente se estima en $2,5 \times 10^{10}$ para el año 2020 y más de



1×10^{11} dispositivos para el año 2050 [2].

El almacenamiento y procesamiento de los datos generados por estos sensores no representa un punto de dificultad, si lo es la gestión, organización y publicación de cada dato generado por los nodos en alguna aplicación en particular.

A pesar de los esfuerzos de normalización y que las alianzas industriales han hecho un gran trabajo en los últimos años para reducir el número de protocolos y establecer un nuevo nivel de servicios ([3], [4], [5], [6], [7]), como también en software de arquitecturas generalizada [8], distantes nos encontramos de un punto en el que todos los dispositivos observen un protocolo común [9]. Por otro lado, la estandarización de protocolos con el objeto de que las diferentes redes de sensores desplegadas hasta el momento presente una aceptable compatibilidad, implicaría una drástica modificación en la arquitectura de los nodos de las redes, es decir en la estructura operacional fundamental del sistema que lo conforman.

En la búsqueda de simplificar el método reduciendo el tiempo en el proceso de comunicación “máquina-máquina” (M2M), al mismo tiempo optimizar el ancho de banda utilizado en publicar los datos generados por cada sensor, se encuentra el protocolo Message Queuing Telemetry Transport (MQTT) [10]. MQTT fue inventado por IBM y Eurotech como un protocolo de conectividad extremadamente liviano del modelo publicidad/suscripción, basado en TCP/IP. Su publicación y suscripción se organizan sobre la base de la noción de “tema”. MQTT permite a los dispositivos abrir una conexión, la mantiene abierta utilizando poca energía, y entrega los mensajes con pequeña sobrecarga transporte. Es utilizado en implementaciones con dispositivos de limitado ancho de banda, satelitales o dial-up con proveedores de servicio de salud, y diversas aplicaciones de domótica, siendo muy útil en la comunicación con dispositivos de baja potencia. MQTT actualmente está siendo normalizado por OASIS [11].

Características de mencionado protocolo son; a) simplicidad: ya que es un protocolo que ocupa un espacio de memoria del orden de los kilobytes y de bajo consumo, b) eficiencia, ya que los encabezados son cortos y evita el polling de la red c) orientación a eventos (cf. [12]).

En MQTT, una parte esencial es el bróker, quien alberga los tópicos a los cuales los clientes pueden suscribirse para publicar, siendo el broker el encargado de difundir los mensajes.

El mecanismo de funcionamiento del protocolo es presentado en la Figura 1. El “CLIENTE A” publica en el tópico “temperatura” el valor “22,5”, luego es el broker quien se encarga de replicar el valor publicado por “A” hacia los clientes “B” y “C”. Utilizando el mismo esquema el rol de “publicador” puede ser ocupado por los restantes clientes conectados al bróker.

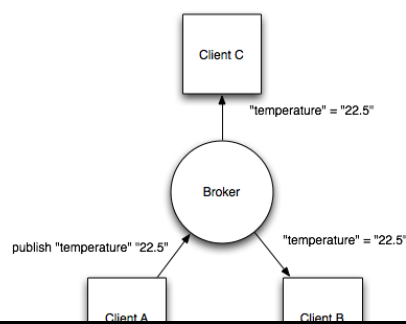




Fig. 1: Comunicación entre clientes en MQTT

En la Universidad Nacional de Misiones se ha desplegado una red de características híbrida, de tal manera que coexistan las redes cableadas con la red de sensores inalámbricos (WSN) existente. Para lograr que las lecturas de los diferentes sensores puedan ser puestas a disposición en forma remota, se implementaron archivos de procesamiento por lotes desarrollados en lenguaje Python que se conecta en modo cliente a un bróker del protocolo MQTT.

Este script es quien se encarga de coleccionar los datos generados por los nodos de la WSN y publicarlos en diferentes tópicos existentes sobre diferentes temas en diversos servidores públicos y/o privados, como Mosquitto.org (cf. [13]). Mosquitto escucha en diferentes puertos a MQTT, en el puerto 1883 conexiones sin cifrar, 8883 conexiones cifradas, 8884 requiere al cliente certificado de cifrado, 8080 Websockets sin cifrar, y 8081 Websockets cifrado. Los puertos cifrados TLS¹ v1.2, v1.1 o v1.0 son gestionados con certificados x509. El servidor provee la autoridad de certificación para verificar la conexión establecida.

¹ TLS=Transport Layer Security

Trabajos Relacionados

Chen y Lin (cf. [14]) proponen un método para integrar MQTT con la arquitectura ETSI M2M a través de una nueva función de red llamada MQTT proxy. El proxy MQTT, por un lado, actúa como un agente de MQTT para los clientes MQTT. Mientras que por otro lado, sirve como una aplicación de puerta de enlace para la interconexión con la arquitectura ETSI compatible M2M, específicamente OpenMTC desarrollado por Fraunhofer FOKUS. Luzuriaga et al. (cf. [15]) considera como los protocolos más utilizados en IoT, y entornos máquina a máquina (M2M) a MQTT, Constrained Application Protocol (CoAP) y Lightweight M2M (LWM2M), dependiendo siempre de la familia de protocolos TCP/IP, considerando su utilización redes cableadas, ya que la familia de protocolos es muy confiable cuando se utilizan en éstas. Esto no se cumple en presencia de conexiones intermitentes como en los enlaces inalámbricos. Los autores proponen mejorar MQTT con énfasis en escenarios móviles, volviendo al sistema más inmune a los cambios en el punto de unión de los dispositivos móviles, evitando que los desarrolladores de servicios para IoT tengan en cuenta explícitamente esta cuestión; mostrando que la propuesta basada en un búfer intermedio, garantiza que no haya pérdida de información durante los períodos de hand-off debido a la movilidad del nodo;

Cendón (cf. [16]) concluye que de acuerdo a como se planifican y se integran las implementaciones actuales M2M, se puede afirmar que en la mayoría de los casos, se toman tres enfoques diferentes en la presentación de la arquitectura de interconexión: a) la arquitectura de tres niveles con dispositivos finales no IP b) arquitectura de dos niveles con los dispositivos finales habilitados para arquitectura IP c) Arquitectura de dos niveles con los dispositivos finales



que no son IP; poniendo de manifiesto un panorama tecnológico heterogéneo en donde se proponen arquitecturas distintas con y sin dispositivos capaces de manejar el protocolo IP pero con una capa física en común. Este tipo de arquitectura es empleada usualmente en el despliegue de redes de sensores que requieren un gran número de nodos manteniendo el costo bajo.

Metodología

El trabajo se ha desarrollado considerando a los siguientes objetivos:

- Lograr una integración flexible de la red de sensores/actuadores ya existente al concepto de internet de las cosas
- Propender a una comunicación sin complicaciones entre los actores intervinientes.
- Implementar los protocolos simple y rápidamente.
- Desarrollar y aplicar tecnología de bajo costo
- Evitar modificaciones sustantivas en la infraestructura de red existente.

En diversos trabajos en la UNaM se ha avanzado sobre ensayos e implementaciones de sistemas electrónicos de captura de información en diversos entornos, tal como un sistema de generación fotovoltaica integrado a la red de distribución eléctrica [17], entre otros.

Una debilidad del esquema actual de la operación de estos sistemas, se presenta al momento de sociabilizar datos en ambientes como el WWW, considerando la selección de la aplicación, la latencia de los vínculos, y la complejidad inherente del método seleccionado. El método implementado como norma para comandar actuadores de forma remota, consiste en la modificación de estado de registros en tablas de una base de datos. Leyendo y recuperando información de las tablas, un script es encargado de notificar la modificación del registro correspondiente, enviando el comando de control al nodo para que ejecute la acción determinada.

La red de sensores desplegada actualmente en la Facultad de Ciencias Exactas de la UNaM ha sido implementada en una topología estrella, donde todos los nodos sensores envían las lecturas correspondientes al concentrador. En el nodo concentrador se ha instalado, por su bajo costo de adquisición y ubicuidad de sus componentes, en una computadora de bajo costo Raspberry Pi [18], corriendo LAMP (Linux, Apache, MySQL y PHP-Python-Perl), estableciendo de ésta manera un servidor WWW. El nodo concentrador se vincula a la computadora a través de un adaptador USB como se muestra en la Figura 2.

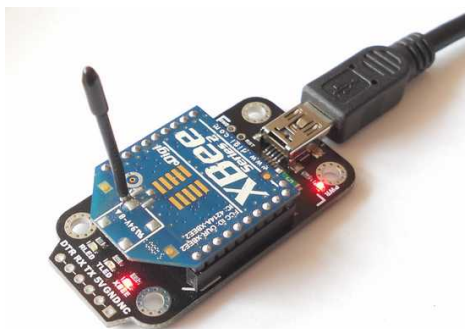




Fig. 2 - Nodo concentrador de la WSN y Adaptador USB

Cada nodo remoto de la WSN está formado por: a) microcontrolador de bajo costo basado en una arquitectura de 8 bits, b) módulo de comunicaciones XBEE PRO HP (902-928 MHz), c) sensores y d) actuadores. Esos nodos capturan información del medio a partir de diversos sensores a saber: temperatura, humedad e iluminación, o también recibiendo órdenes por parte del gateway/concentrador para cambiar el estado de algún actuador. Los actuadores implementados son relés de estado sólido como el presentado en la figura 3 y esquematizados en la figura 4 por una llave tipo ON/OFF (ver nodo 2) con el objeto de comandar cargas de corriente alterna.

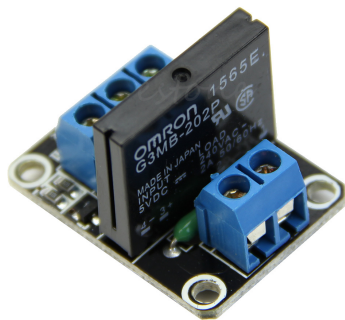


Fig.3 - Relé de estado sólido utilizado como actuador

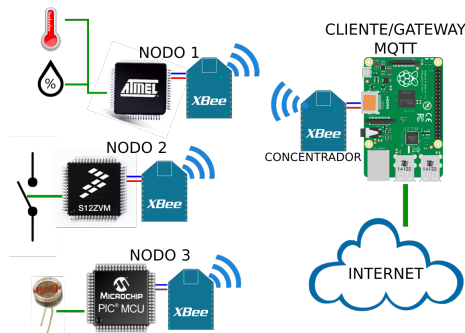


Fig. 4 - Esquema de la WSN y Gateway MQTT

Implementación de un Cliente MQTT

Los avances alcanzados en la tecnología de las redes inalámbricas, nos han colocado en la puerta de una nueva era, en la cual pequeños equipos inalámbricos nos proveerán acceso a la información en cualquier momento y en cualquier lugar, permitiéndonos asimismo participar activamente en la creación de “espacios inteligentes”. Hoy se está experimentando en las TICs los efectos del vasto número de objetos heterogéneos existentes en una red descentralizada, fundamentalmente en los efectos de escala en la interacción entre hardware, software, algoritmos y datos donde la mayoría de los fenómenos ocurren sorpresivamente y no como una definición del diseño.



Internet de las Cosas es el entorno inteligente que conecta los dispositivos a Internet. Estos entornos presentan ciertos requisitos acerca de la naturaleza y de la gestión de los nodos involucrados, razón por lo cual para satisfacerlos, el ambiente precisa indefectiblemente de un nodo pasarela/gateway que pueda administrarlos.

Una puerta de enlace compatible, definida como parte de Internet de las Cosas, debe ser compatible con diversos protocolos de acceso, ya sean definidos por IP (Ethernet, Wifi, etc.) o no (ZigBee, Bluetooth, etc.). Sin embargo, los datos primarios generados por estos dispositivos poseen una heterogeneidad en la estructura de datos, funcionalidad, etc.; razón por la cual la reutilización de los datos generados por el sensor y al acceso a los dispositivos es una tarea difícil. Dado que el hardware sobre el cual se basan los nodos de la red actual no posee capacidad de manejar en forma directa la pila TCP/IP, surge como propuesta utilizar el mismo nodo concentrador de la figura 4 como agente intermediario de esta tarea. Para llevar esto a cabo se ha desarrollado un script en un lenguaje de alto nivel como Python, que gracias al vasto soporte existente y clases disponibles permite gran interacción tanto con periféricos existentes en la computadora (por ej. puerto serie), como así también con aplicaciones web.

El script `gateway_MQTT.py` cumple 3 funciones claramente diferenciables: i) compilar la información de los nodos sensores distribuidos, ii) publicar las lecturas de cada nodo en un tópico en particular a los cuales el cliente se encuentra suscripto y iii) recibir a través de las suscripciones a los diferentes tópicos los comandos para los actuadores.

Una función encargada de compilar la información envía de a un nodo por vez una solicitud de lectura a cada nodo empleando la clase "pyserial" de Python [19], conectando luego a un bróker determinado utilizando la clase "paho-mqtt" mediante el método `connect()` para la vinculación. Establecida la conexión, el script hace uso de los métodos `subscribe()` y `publish()` para suscribirse a diferentes tópicos y publicar en estos, los valores generados por los nodos de la WSN.

Si existiesen nodos con algún actuador, el coordinador/gateway es quien escucha los mensajes publicados en un tópico en particular y envía los comandos a estos nodos actuadores para actuar en consecuencia.

Resultados y Discusión

En este trabajo se ha definido la implementación de un cliente del protocolo MQTT como *gateway* de una red de sensores inalámbricos existente en la Universidad Nacional de Misiones. La solución propuesta consistió en un script de lenguaje Python corriendo en una computadora de bajo costo Raspberry Pi.

La implementación e integración del protocolo MQTT a la WSN existente se realizó de forma exitosa. El hardware de los nodos sensores no fue modificado por lo que no existió erogación alguna. El desarrollo del script se ha mantenido lo más simple, claro y escalable posible.

Si bien sería válido proponer una migración total de la red de sensores existente a una donde cada uno de los nodos sea un cliente del bróker por medio de las alternativas existentes (Raspberry, Beaglebone, Intel Galileo), hoy en día representa un costo/nodo relativamente elevado en nuestro país. Sin embargo, mantener los nodos remotos basados en una tecnología de menores prestaciones y hacer que los mismos converjan a través de otras formas de comunicación a un coordinador que presta servicios de enrutamiento IP reduce notablemente los costos.



El mando de actuadores se simplifica considerando que se podría prescindir de una base de datos como punto intermedio entre los nodos de la red y el usuario remoto, permitiendo ubicuidad en la etapa de registro y almacenamiento de los datos en cada tópic.

Conclusiones

Entre las actividades propuestas se considera una migración de la plataforma actual en la que se encuentran basados los nodos. La alternativa a plantear, atractiva y de un costo relativamente bajo, consistiría en migrar el hardware actual a uno basado microcontrolador ESP8266, donde cada nodo de la WSN poseería embebido la pila TCP/IP, de esta manera cada uno de los nodos ocuparía el rol de cliente, pudiendo interactuar forma directa al bróker MQTT.

Referencias

- [1] IETF -Network Working Group, «RFC 2460 - Protocolo Internet, Versión 6 (IPv6),» 1998.
- [2] IBM Institute for Business Value, «Device democracy, saving the future of the internet of things».
- [3] KNX Association, «KNX domótica,» [En línea]. Available: <https://www.knx.org>.
- [4] Open Connectivity Foundation, «New OCF Spearheading Standard Efforts for IoT,» 2016. [En línea]. Available: <https://openconnectivity.org/>.
- [5] Bluetooth SIG, Inc. , «Bluetooth Technology Website,» [En línea]. Available: <https://www.bluetooth.com/>. [Último acceso: 2016].
- [6] W.-F. Alliance, «The worldwide network of companies that brings you Wi-Fi,» 2016. [En línea]. Available: <http://www.wi-fi.org/>.
- [7] The ZigBee Alliance , «Control your World,» 2016. [En línea]. Available: www.zigbee.org.
- [8] The Apache Software Foundation, «Apache River News,» 2016. [En línea]. Available: <https://river.apache.org/>.
- [9] I. Georgievski, V. Degeler, G. Pagani, T. Nguyen, A. Lazovik y M. Aiello, «Optimizing Energy Costs for Offices Connected to the Smart Grid,» IEEE Transactions on Smart Grid, vol. 3, n° 4, pp. 2273-2285, 2012.
- [10] MQTT, «MQTT connectivity protocol,» 2016. [En línea]. Available: <http://mqtt.org>.
- [11] OASIS, «Advancing Open Standards for the Information Society,» 2016. [En línea]. Available: <https://www.oasis-open.org/>.
- [12] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes y M. Mohammadi, «Toward better horizontal integration among IoT services,» IEEE Communications Magazine, vol. 53, n° 9, pp. 72-79, 2015.
- [13] Mosquitto, «test.mosquitto.org,» 2016. [En línea]. Available: test.mosquitto.org.



- [14] H. Chen y F. J. Lin, «Converging MQTT Resources in ETSI Standards Based M2M Platform,» IEEE International Conference on Internet of Things (iThings), and Green Computing and Communications (GreenCom), pp. 292-295, 2014.
- [15] J. E. Luzuriaga, J. Cano, C. Calafate, P. Manzoni, M. Perez y P. Boronat, «Handling mobility in IoT applications using the MQTT protocol,» Internet Technologies and Applications (ITA), pp. 245-250, 2015.
- [16] B. Cendón, Machine-to-machine (M2M) Communications. Architecture, Performance and Applications, vol. 1, C. A. a. M. Dohler, Ed., Elsevier Ltd., 2015.
- [17] M. E. Sosa, L. A. Urbani, G. A. Xánder y G. A. Fernández, «Sistema de Monitoreo online para paneles fotovoltaicos utilizados en generación distribuídas,» de XXVI Congresso Regional de Iniciação Científica e Tecnológica em Engenharia - CRICTE, Alegrete - Brasil, 2014.
- [18] Raspberry Pi Foundation, «Teach, Learn and Make with Raspberry Pi,» 15 Oct 2015. [En línea]. Available: <https://www.raspberrypi.org/>.
- [19] Python, «pyserial 3.1- Python Serial Port Extension,» 2016. [En línea]. Available: <https://goo.gl/hkE6cz>.