Brenda Paola Castillo Torres - A01632227

Jorge Alejandro López Sosa - A01637313

Andres Eduardo Nowak de Anda - A01638430

Roberto López Cisneros - A01637335

**Investigación sobre Protocolo MQTT**

***Protocolo MQTT***

MQTT es un protocolo de comunicación que ha avanzado bastante porque es muy utilizado en el Internet de las Cosas, y también tiene bastantes ventajas porque es un protocolo abierto; además, en los otros protocolos conocidos se necesita una nueva conexión cada que se realiza una nueva comunicación.

El protocolo MQTT utiliza un filtro para los mensajes que son enviados a cada usuario, y esos filtros están basados en temas que se organizan de forma jerárquica (topics). De esa manera, todos aquellos usuarios conectados que se suscriban al topic, recibirán mensajes a través del servidor. Sin embargo, como es MQ, los mensajes se mantendrán en la cola y no se perderán hasta que el usuario no reciba el mensaje.

Mientras que las conexiones se realizan por medio de TCP/IP, el servidor mantendrá un registro de los usuarios conectados. Por default, los dispositivos emplean puertos de comunicación de número 1883, aunque en ocasiones se usa el 8883 si se usa el SSL/TLS.

Sin embargo, no solamente se necesita usuarios, servidor, y puertos para que la conexión funciones, sino que también se necesitan otros paquetes como el CONNECT que contiene toda la información del usuario, y después el servidor responde con un CONNACK que avisa al usuario si la conexión fue aceptada, rechazada, entre otros.

Una vez establecida la conexión se usan paquetes PUBLISH con el topic y se envía al servidor. Por el otro lado, el usuario interesado usará el SUBSCRIBE y UNSUBSCRIBE para suscribirse o retirar suscripción. Después el servidor responde con SUBACK y UNSUBACK si la operación realizada por el usuario es un éxito.

Finalmente, para finalizar la conexión se envía un paquete DISCONNECT, pero si queremos mantener la conexión, los usuarios mandan constantemente el paquete PINGREG que será correspondido con un paquete PINGRESP del servidor.

Como se mencionó anteriormente la función de MQTT, podemos ver que dicho protocolo aporta muchas ventajas, como el asincronismo, sencillez, flexibilidad (uso para diversas aplicaciones), escalabilidad (conecta más y más usuarios), también otras de las ventajas es que es un protocolo que tiene un menor consumo de recursos a comparación de otros protocolos como el HTTP, y en el tema de menor consumo, también tiene un menor consumo de potencia ya que no necesita de un gran ancho de banda.

Aunque, el protocolo MQTT tiene sus ventajas, también tiene sus desventajas y es que dicho protocolo no está protegido, a menos que se utilice el TLS/SSL para una mejor seguridad, también alguna de sus desventajas es que es muy difícil de crear una red MQTT globalmente escalable, y otra desventaja es que tiene ciclos de transmisión más lentos que en comparación de otro protocolo como el CoAP.

***MQTT en modelo de capas***

Es importante mencionar que MQTT se podría encontrar en la segunda capa del modelo de IoT visto durante clase, ya que sería un protocolo que permitiría la conectividad y comunicación de los datos.

Tomando en cuenta el modelo OSI revisado durante el curso, el protocolo MQTT tendría su funcionamiento en el séptimo nivel o capa, es decir en la capa de aplicación que es la que permite acceder a todos los demás servicios de las capas debajo de ella. En esta capa se encuentran los protocolos que permiten a las aplicaciones intercambiar datos entre sí; tomando en cuenta lo mencionado anteriormente, MQTT serviría como un protocolo para el intercambio de datos entre diversas aplicaciones, ya que permite publicar información a los usuarios o aplicaciones que se encuentran suscritos a distintos tópicos; a partir de esta información, se puede actuar y enviar alguna respuesta de regreso o llevar a cabo un análisis de estos datos.

***Publish-Subscribe - Modelos de comunicación***

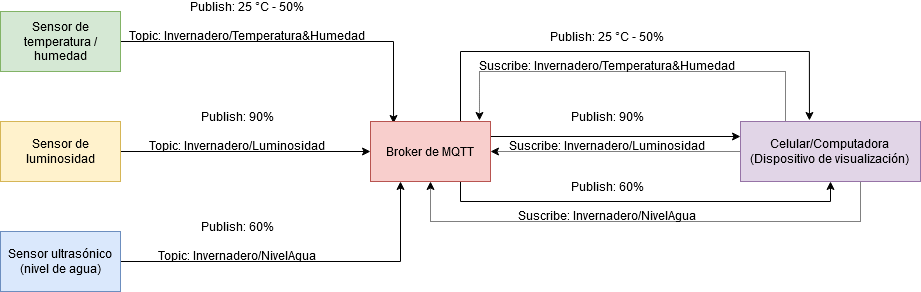
D2D: En este caso uno de los dispositivos sería el que se suscribe y el otro es el que va a estar publicando los datos. Debido a que se trata de un modelo en el que no se tienen gateways de por medio, la comunicación entre el dispositivo que publica y el que se suscribe sería directa.

D2C: En esta parte el dispositivo / sensor sería el que hace publish ya que es el que va a mandar de manera constante los datos a la nube; mientras que la nube sería el que se subscribe para recibir los datos, ya que es el que los recibe.

D2G: En este caso el dispositivo / sensor es el publisher ya que manda los datos directamente al gateway, el gateway sería broker encargado de mandar los datos a la nube o otra computadora (se podría considerar a la nube o al otro dispositivo que recibe los datos como el que lleva a cabo el suscribe porque es a donde llegan los datos).

Back-End Data-Sharing Model: En este caso se tiene un publisher que serían los sensores y hay un subscriber que sería una nube; esta puede mandar los datos a otras nubes o otros dispositivos autorizados que lo requieran por el medio de uso como una api, es decir, que existen varios publishers y subscribers en la nube.

***Cultivo hidropónico con MQTT***



En este caso, se tienen 3 sensores que mandarán datos al broker MQTT, estos son el sensor de temperatura y humedad, el de luminosidad y el ultrasónico que mide el nivel de agua; cada uno de los sensores publica los datos a un tópico específico.

Una vez que llegan al broker, éste publica los datos a los dispositivos que tengan una suscripción a los respectivos tópicos disponibles. En este caso, debido a que se quiere que se tenga toda la información de todos los sensores en un mismo dispositivo para que el usuario pueda acceder a ellos, el dispositivo está suscrito a los 3 tópicos; sin embargo, en el caso de ser requerido por otro usuario, se pueden agregar otros dispositivos que solo se encuentren suscritos a uno o dos tópicos, por lo que solamente tendrán acceso a dichos datos.

***Adaptaciones Wifi a Sigfox***

En el caso de la propuesta del reto se utiliza el protocolo de acceso a la red Wifi porque es uno de los más simples de implementar. Sin embargo, en el caso del control de un cultivo hidropónico, es importante tomar en cuenta que se tienen 2 usuarios principales que podrían utilizar la aplicación. En el caso de las personas que tienen su propio huerto hidropónico en casa es adecuado el uso de Wifi, ya que tiene pocas implicaciones que se deben llevar a cabo para que el usuario pueda acceder a la información relacionada con sus cultivos.

Por otro lado, para los usuarios que cuentan con invernaderos hidropónicos de gran tamaño o incluso para los agricultores, el uso de wifi no sería el más adecuado por diversas razones incluido su limitado alcance, costo, razón de transferencia de datos y consumo de energía. Ante esta situación, se optó por una mejor alternativa, que se trata de Sigfox.

Sigfox es una red que permite la conectividad al Internet para los dispositivos del Internet de las Cosas, sus principales características son que es muy barato de implementar, tiene un alcance de hasta 50 km en el campo y 10 km en la ciudad, tiene una baja frecuencia de transferencia de datos de hasta 140 mensajes por día y el consumo de energía de los dispositivos es muy bajo, se pueden transferir hasta 600 bits por segundo y utiliza 200 khz de la red y trabaja en frecuencia de 896 a 928 MHz.

La manera en la que funciona es muy similar al Wifi, sin embargo, se requieren de algunas adecuaciones para poder utilizarse. Los dispositivos requieren la contratación de la red Sigfox y conectarse a ella para enviar datos a un gateway Sigfox. En el caso del nodeMCU se puede reemplazar por el módulo WISOL RCZ2 que tiene la capacidad de conectarse a redes Sigfox. Los datos son transmitidos a través de un canal llamado UNB (Ultra Narrow Band) que está diseñado para funcionar con una baja transferencia de datos.

Los gateways de Sigfox trabajan utilizando DPSK para mandar los datos a la nube, esto hace que se convierta y analice la señal para eliminar cualquier tipo de interferencia.

En resumen, la única adecuación que se requeriría llevar a cabo sería el cambio de la red Wifi por Sigfox y el funcionamiento adicional sería muy similar.

**REFERENCIAS**

Isaac. (2020). MQTT: un protocolo abierto de red y su importancia en el IoT. *Hardware Libre*. Recuperado de: <https://www.hwlibre.com/mqtt/>

Michaslki, T. (2017). Explaining Sigfox. *Ubidots.* Recuperado de: <https://ubidots.com/blog/explaining-sigfox/>

Ray, B. (2018). SigFox Vs. LoRa: A Comparison Between Technologies & Business Models. *Link Labs.* Recuperado de: <https://www.link-labs.com/blog/sigfox-vs-lora>

Tecnología Sigfox: Internet de las cosas. (s.f). *Dset.* Recuperado de: <http://productos-iot.com/sigfox-3/>

The Four Internet of Things Connectivity Models Explained. (2016). *Inet Services Cloud.* Recuperado de: <http://www.inetservicescloud.com/the-four-internet-of-things-connectivity-models-explained/>