

# Protocolos en IoT en NodeRED

Taller de Nuevas Tecnologías (IF017)

Nahuel Defossé

# Protocolos de IoT

# División por funcionalidad

En Internet de las Cosas (IoT) existe variedad de protocolos que pueden ser analizados desde diferentes perspectivas o necesidades:

- ▶ Infraestructura (ej: 6LowPAN, IPv4/IPv6, RPL)
- ▶ Identificación (ej: EPC, uCode, IPv6, URIs)
- ▶ Comunicación y Transporte (ej: Wifi, Bluetooth, LPWAN)
- ▶ Descubrimiento (ej: Physical Web, mDNS, DNS-SD)
- ▶ Protocolos de Datos (ej: MQTT, CoAP, AMQP, Websocket)
- ▶ Administración de dispositivos (ej: TR-069, OMA-DM)
- ▶ Semánticos (ej: JSON-LD, Web Thing Model)
- ▶ Frameworks multicapa (ej: Alljoyn, IoTivity, Weave, Homekit)

# Modelo de referencia de Estándares Abiertos



Or in industry speak ...



## Open Standards Reference Model

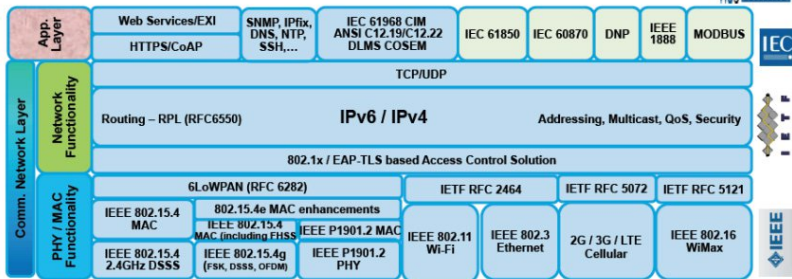


Figura1: Estándares Abiertos

Según Prof.David E. Culler, un ejemplo de la evolución hacia las redes de área amplia de baja velocidad se puede ver en The Internet of Every Thing - steps toward sustainability CWSN

# Ancho de banda vs alcance

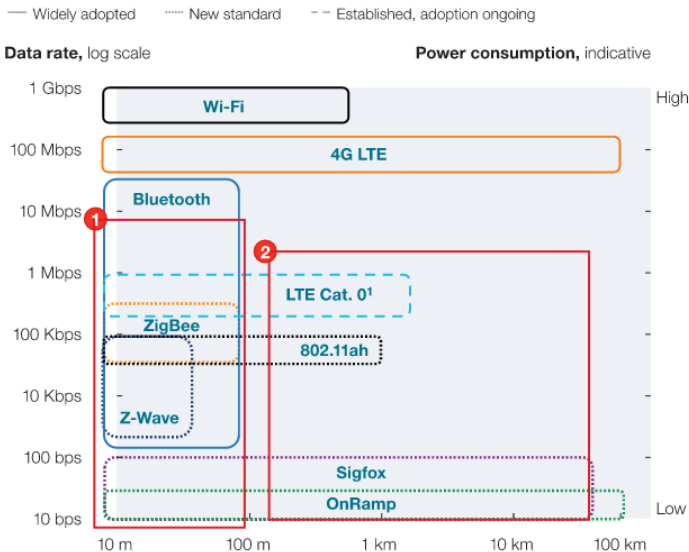


Figura2: Protocolos de Comunicación

# Dispositivos WiFi

- ▶ Conectados a WiFi (802.11b/g/n)
- ▶ Gran ancho de banda.
- ▶ Conexión directa a internet.
- ▶ Disponibilidad de energía.
- ▶ Lámparas inteligentes. Ej: Hue
- ▶ Termostatos inteligentes: Ej: Nest
- ▶ Colchones inteligentes: Ej: Luna



Figura3: Dispositivos WiFi



# Ancho de banda Bluetooth

- ▶ Conectados por batería
- ▶ Se conectan con algún dispositivo WiFi de manera eventual o para completar su funcionalidad
- ▶ Registro de actividad/salud.
- ▶ Localizadores (ej: tTile).
- ▶ Relojes inteligentes.



Figura4: Dispositivos Bluetooth

# Dispositivos LTE/4G

- ▶ Disponibilidad de energía superior a Bluetooth
- ▶ Dispersión geográfica media
- ▶ Vehículos
- ▶ Cestos de basura que notifican cuando están llenos

# LoRa

## Características generales

- ▶ Low Power Wide Area Network (LPWAN) patentado por Semtech.
- ▶ Velocidades de 250bps a 50kbps.
- ▶ Frecuencias no licenciadas, USA 915MHz, Europa 868MHz, Asia 433MHz (máxima capacidad de emisión ~25mW).
- ▶ Bajo poder, 12 bytes por mensaje, 144 mensajes por hora.
- ▶ Link Budget de 154dB, vs 130 de LTE
- ▶ Esto quiere decir que en situaciones ideales podría llegar a 1000Km vs 100Km (sin contar atenuaciones de antena u objetos en línea de visión) <sup>1</sup>.
- ▶ Sensores con MCU integrado (ej: DRF1276DM <sup>2</sup>)

---

<sup>1</sup><http://en.jirous.com/calculation-wifi>

<sup>2</sup><https://lora-alliance.org/member-directory>

## Características generales (cont)

- ▶ Gran soporte del estándar <sup>3</sup>
- ▶ Proveedores de comunicaciones instalan antenas que permiten suscribirse al servicio
- ▶ Existe una red dónde se anuncian los proveedores de servicio de forma gratuita:



# Gateways

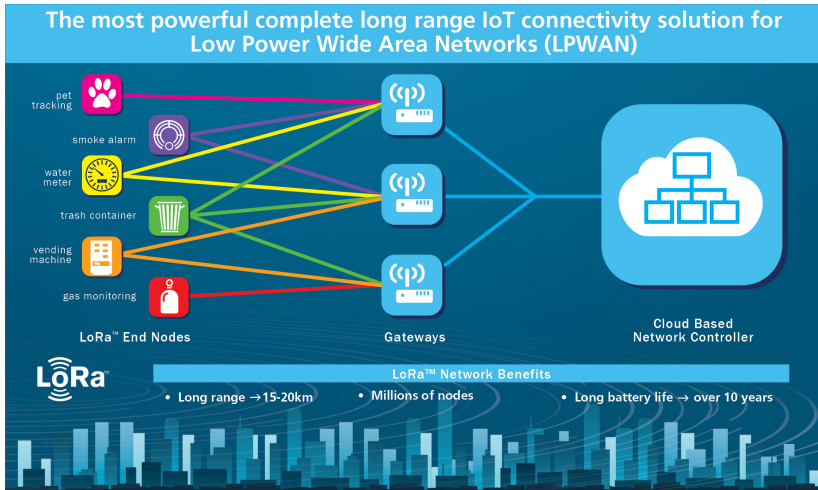


Figura6: La comunicación a través de Gateways

## Características generales

- ▶ Estándar propietario, creado en Francia.
- ▶ Una red celular, para cosas, creado en el 2009.
- ▶ Paquetes de datos pequeños.
- ▶ Bajo consumo de energía (duración de hasta 10 años).
- ▶ Precio muy inferior a redes de celular.
- ▶ Para cubrir la superficie de toda Francia, se necesitaron 1200 torres (vs 50000 para cobertura común de 3G/4G).
- ▶ 12 bytes/mensaje, 140 msg/día.



Figura7: SigFox y sus Gateways



# Cobertura SigFox

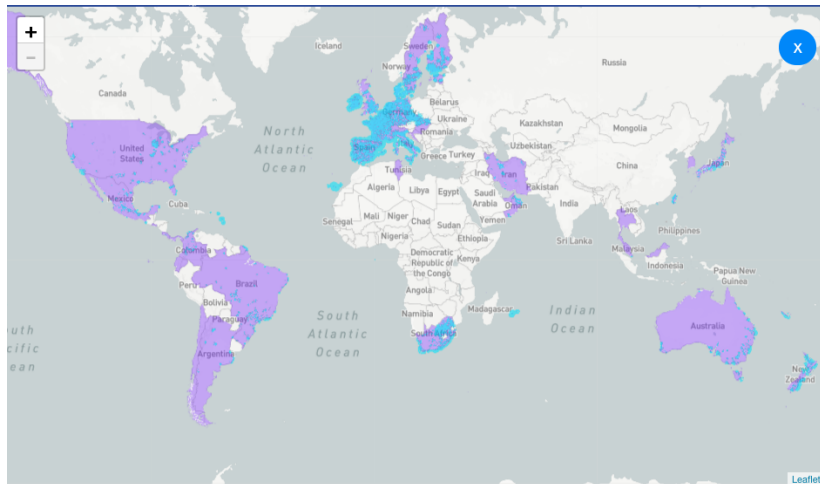


Figura8: Cobertura en el mundo

# MQTT

MQTT es un protocolo publicador subscriptor de envío de mensajes, basado en *broker* (MoM).

Significa Message Queue Telemetry Transport y esta pensado para **bajo ancho de banda, alta latencia, enlaces poco confiables**, de **alto costo**, pero al estar basado en TCP, no es tan minimalista como **LoRa** o **SigFox** (también en diferente capa OSI).

Los mensajes son de bajo *overhead*, 4 bytes. Actualmente en su versión 5, aunque su especificación más popular es MQTT 3.1.1.

# MESSAGING STANDARDS

*What messaging protocol(s) do you use for your IoT solution?*

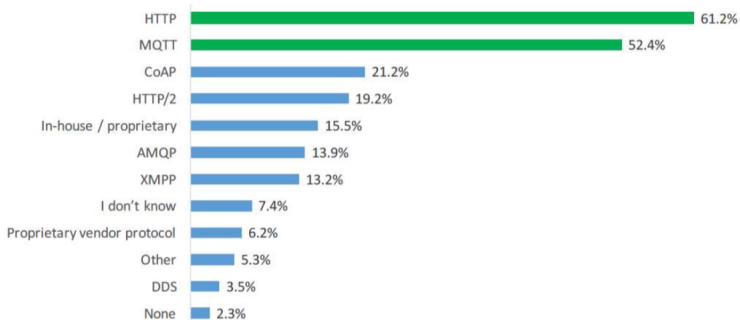


Figura9: Uso de los protocolos según la Eclipse Foundation

# Comunicación Broker

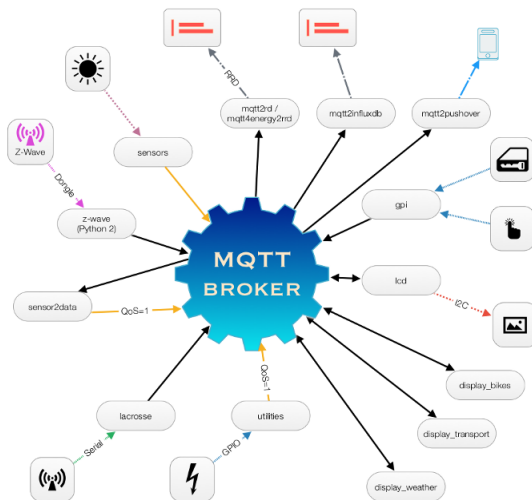


Figura10: Esquema de funcionamiento de MQTT

# Tipos de mensajes

Table 2.1 - Control packet types

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Client request to connect to Server
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server or Server to Client	Publish message
PUBACK	4	Client to Server or Server to Client	Publish acknowledgment
PUBREC	5	Client to Server or Server to Client	Publish received (assured delivery part 1)
PUBREL	6	Client to Server or Server to Client	Publish release (assured delivery part 2)
PUBCOMP	7	Client to Server or Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	8	Client to Server	Client subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgment
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server	Client is disconnecting
Reserved	15	Forbidden	Reserved

## Atributos de la conexión

- ▶ QoS
- ▶ Última voluntad (*last will*)
- ▶ Voluntad Inicial (*birth*)
- ▶ Intervalo entre pings (tiempo de vida)
- ▶ Identificador de cliente
- ▶ Clean Session

## Atributos del mensaje

- ▶ Retenido
- ▶ QoS
- ▶ Como máximo una vez (0)
- ▶ Al menos una vez (1)
- ▶ Exactamente una vez (2)

## Transportes de MQTT

- ▶ Transporte TCP, puerto típico 1883
- ▶ TLS en el puerto 8883.
- ▶ Transporte WebSocket
- ▶ MQTT-SN (Sensor Network), funciona cambiando tópicos.

## Semántica de los Tópicos

- ▶ Todos los tópicos deben tener al menos un caracter
- ▶ Tanto los tópicos como los filtros, son sensibles a mayúsculas
- ▶ Se puede incluir el espacio
- ▶ Un carácter / al comienzo o al final, no es opcional y cambia el tópico.
- ▶ Es válido el tópico /
- ▶ Son cadenas UTF-8 y no están permitidos los caracteres NULL.
- ▶ Por convención, se utiliza un carácter / como iniciador y separador.



## Filtros

En el momento de la subscripción existen dos tipos de filtros:

- ▶ El carácter +, es un comodín, **de un solo nivel**. Ej:  
/casa+/termometro
- ▶ El carácter # es un comodín **multinivel**. Ej: /casa/#.
- ▶ Los tópicos que contienen el caracter \$ están reservados.

## Implementaciones

- ▶ Mosquitto (C)
- ▶ HiveMQ (Java)
- ▶ Mosca (JS)
- ▶ Apache ActiveMQ y Apollo (Java)
- ▶ emqttd (Erlang)
- ▶ CloudMQTT
- ▶ IBM Integration Bus
- ▶ RabbitMQ

Lista completa en

<https://github.com/mqtt/mqtt.github.io/wiki/servers>

## Autenticación

La autenticación se delega al Broker, en el caso de Mosquitto, JP Mens escribió el plugin mosquitto-auth-plug.

Mecanismos soportados:

- ▶ MySQL
- ▶ PostgreSQL
- ▶ CDB
- ▶ SQLite3
- ▶ Redis key/value store
- ▶ TLS PSK
- ▶ LDAP
- ▶ HTTP (custom HTTP API)
- ▶ JWT
- ▶ MongoDB
- ▶ Archivos

## Bridges (puentes)

Los puentes permiten conectar Brokers de manera que los mensajes publicados en uno, son retransmitidos a otro(s). De esta manera un Broker local puede transmitir a brokers remotos mensajes y viceversa.

Las ofertas de Amazon, Google y Microsoft para IoT, incluyen Brokers que pueden configurarse de esta forma. Es mandatorio en estos casos la utilización de TLS.

## Comunicación entre Brokers

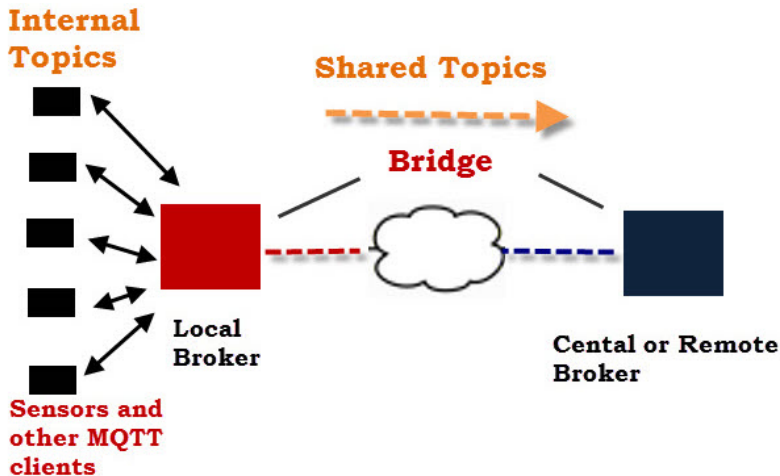


Figura11: Puente entre Brokers

## Hardware Compatible

- ▶ Arduino YUN o Arduino UNO + ESP8266
- ▶ Intel Edison y Galileo
- ▶ Placas basadas en ESP8266 como WeMos, NodeMCU programadas con Arduino
- ▶ RaspberryPI (y similares, OrangePI, BananaPi, etc)

## ¿Dónde seguir?

- ▶ Especificación del standard 3.1.1
- ▶ Cheatsheet de MQTT
- ▶ Wiki de la Comunidad
- ▶ Ejemplo de utilización en Hardware