

Máster en Tecnologías de Análisis de Datos Masivos: BIG DATA

Internet de las Cosas en el Contexto de Big Data

PROTOCOLOS DE COMUNICACIONES EN IoT

Juan Antonio Martínez juanantonio@um.es
Luis Bernal Escobedo Luis.bernal@um.es

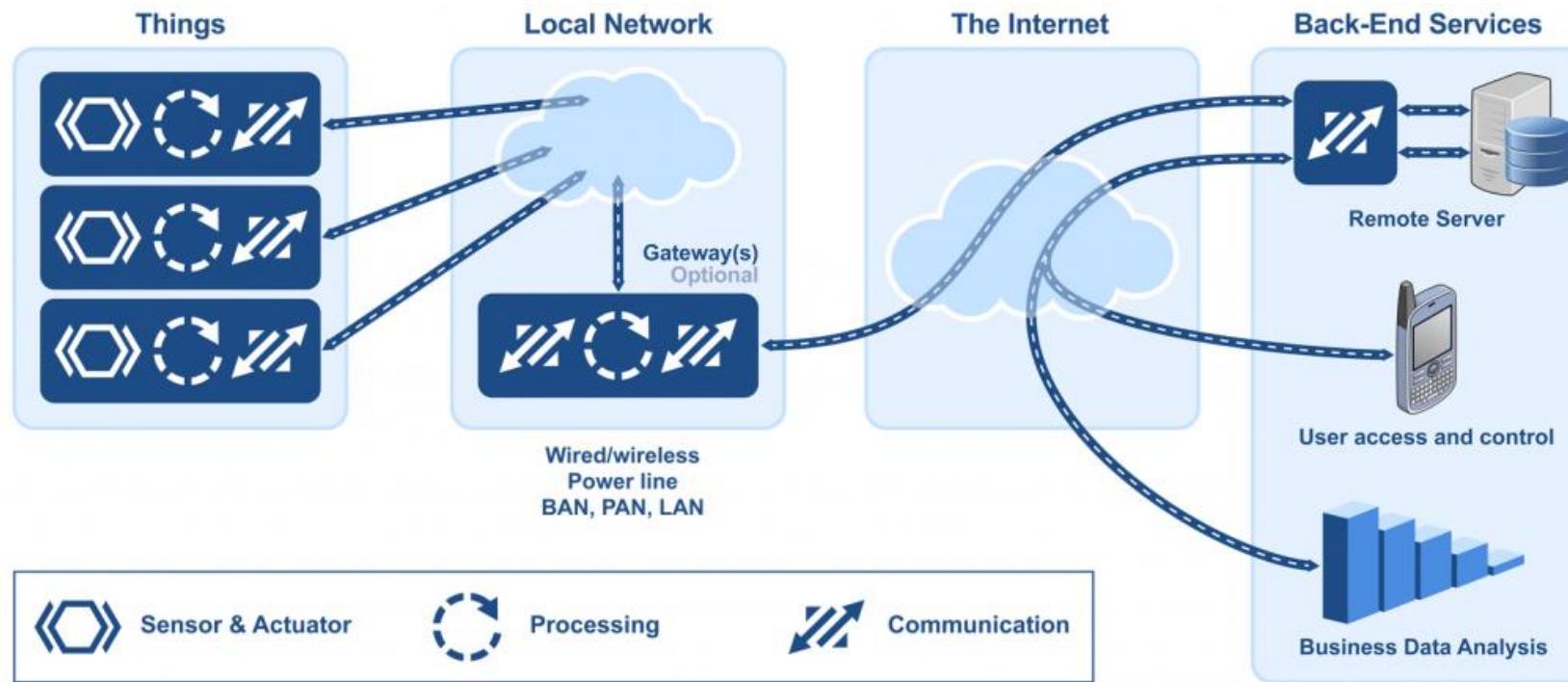


Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

Introducción

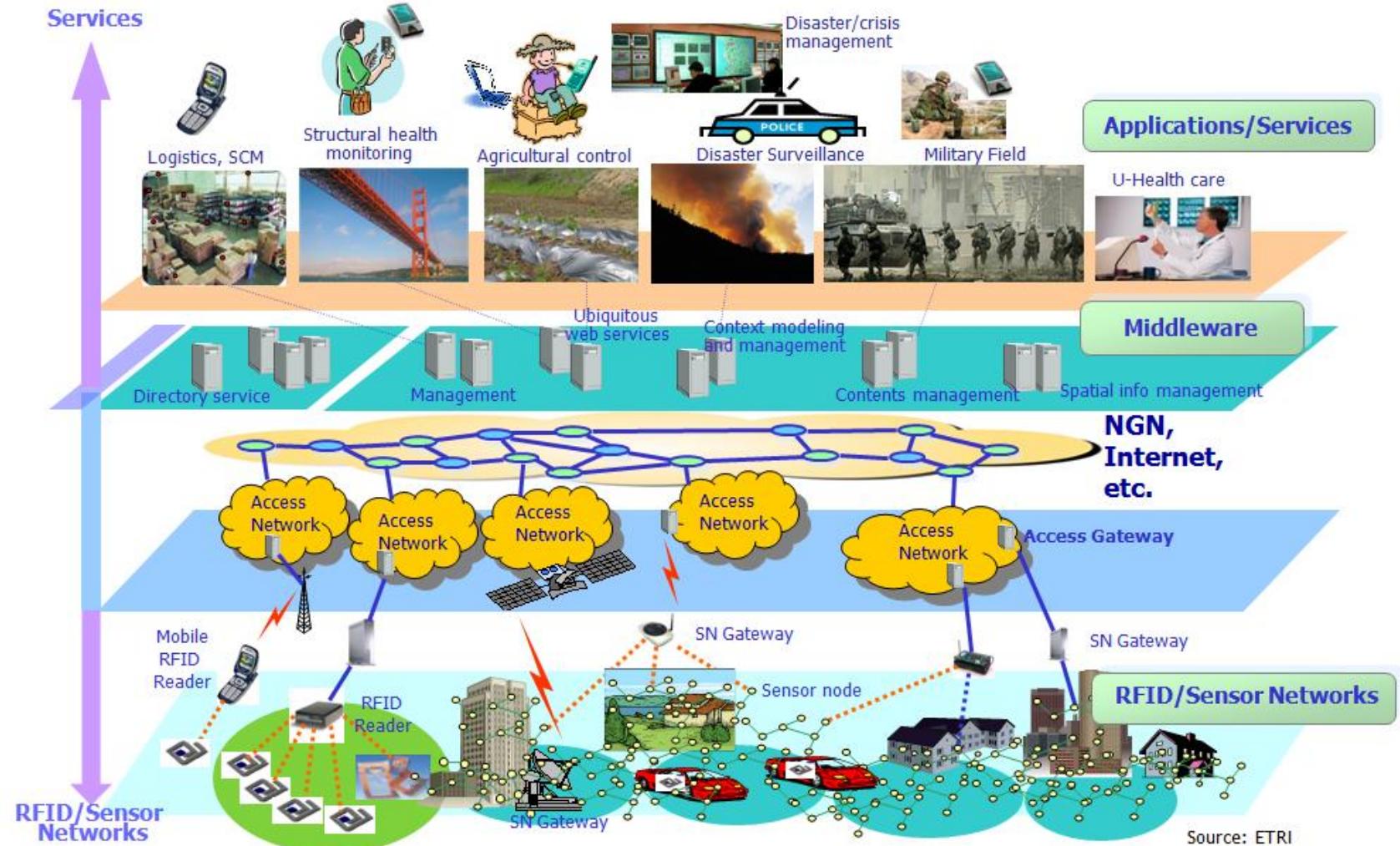
Arquitectura de un Sistema IoT



Introducción

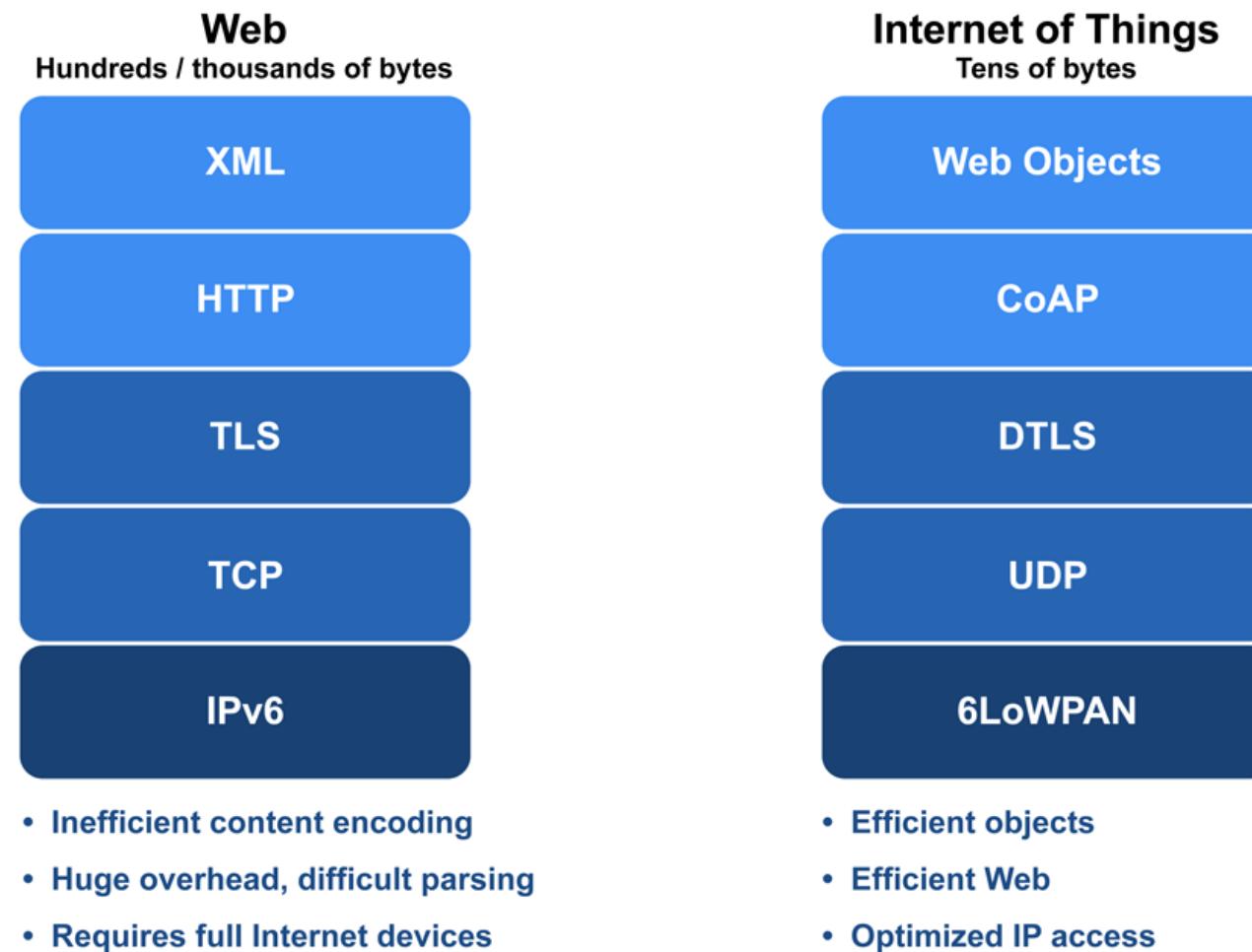
- Ecosistema muy heterogéneo: redes en entornos adversos (ej. industriales), ancho de banda variable, aplicaciones tolerantes a retardos o no, etc.
- Escalabilidad: billones de dispositivos
- Optimización: dispositivos limitados en recursos
- Comunicaciones D2I y D2D
- **IPv6** para direccionamiento global (o no)
- Multitud de protocolos:
 - Nivel enlace y red:
 - **802.15.4**
 - 6LoWPAN
 - Zigbee
 - **LoRaWAN**
 - Bluetooth y **BLE**
 - **Sigfox**
 - WiFi y **HaLow**
 - CAT-M1 y **NB-IoT**
 - Weightless
 - Nivel de aplicación
 - Constrained Application Protocol (**CoAP**)
 - MQ Telemetry Transport (**MQTT**)
 - LWM2M
 - REST API
 - **Protobuf**
 - XMPP (Extensible Messaging and Presence Protocol)





71

Introducción



IoT Estandarización

- ▶ IETF (Internet Engineering Task Force)
 - ▶ 6LoWPAN Working Group (IPv6 global)
 - ▶ CoRE WG (Rest for IoT, CoAP, Resource Directory)
 - ▶ ROLL - Routing Over Low-power and Lossy networks
- ▶ OMA (Open Mobile Alliance)
 - ▶ Lightweight M2M (LWM2M), basado en CoAP, DTLS, REST
 - ▶ NGSI, context management
- ▶ ETSI / OneM2M
 - ▶ Estandarización en comunicaciones M2M, OneM2M, SmartM2M, NGSI-LD, CoAP, HTTP binding...
- ▶ W3C (World Wide Web Consortium)
 - ▶ Efficient XML Interexchange (EXI), estandarización
- ▶ ZigBee Alliance
- ▶ IEEE (Institute of Electrical and Electronics Engineers)
 - ▶ IEEE 802.XX.YY
- ▶ DASH7 Alliance
 - ▶ Dash7
- ▶ OASIS (Organization for the Advancement of Structured Information Standards)
- ▶ MQTT (Message Queuing Telemetry Transport)



IPv6

- ▶ El modelo actual de Internet basada en **IPv4 sufre de importantes limitaciones**
 - Agotamiento del espacio de direcciones (la única razón “poderosa” para impulsar un cambio)
 - Encaminamiento poco escalable
 - Soporte limitado a la movilidad
 - Soporte limitado a la seguridad
- **Ventajas**
 - ▶ Mayor rango de direcciones (128 bits) `3ffe:3328:4:3:250:4ff:fe5c:b3f4`
 - ▶ Jerarquía estructurada para disminuir tamaño de tablas de enrutamiento. Desacopla prefijo (64bits) del identificador del host
 - ▶ Mecanismos de auto-configuración
 - ▶ Mejora en el formato de la cabecera e identificación de flujos
 - ▶ Mejor soporte de opciones y extensiones

Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

IEEE 802.15.4

- ▶ Estándar muy empleado de redes inalámbricas de área personal (WPAN). 1^a versión: 2003, siguen versiones de 2006, 2007 y 2009
- ▶ Utilizado en redes domésticas, control industrial, automatización edificios, WSN... → corto alcance
- ▶ Define control físico y control de acceso al medio (MAC)
- ▶ Bandas definidas: 868 MHz (EU), 915 MHz, 2.4 GHz
- ▶ Tasas bajas de transferencia: 20kbps - 250kbps
- ▶ Baja potencia de emisión: 0.5-100 mW
- ▶ Rango de entre 10m a varios km
- ▶ Modulaciones varias: DSSS, GFSK, QPSK

Physical (L1)
IEEE 802.15.4
868/915 MHz

Data Link (L2)
IEEE 802.15.4 MAC

Upper Layer Stack

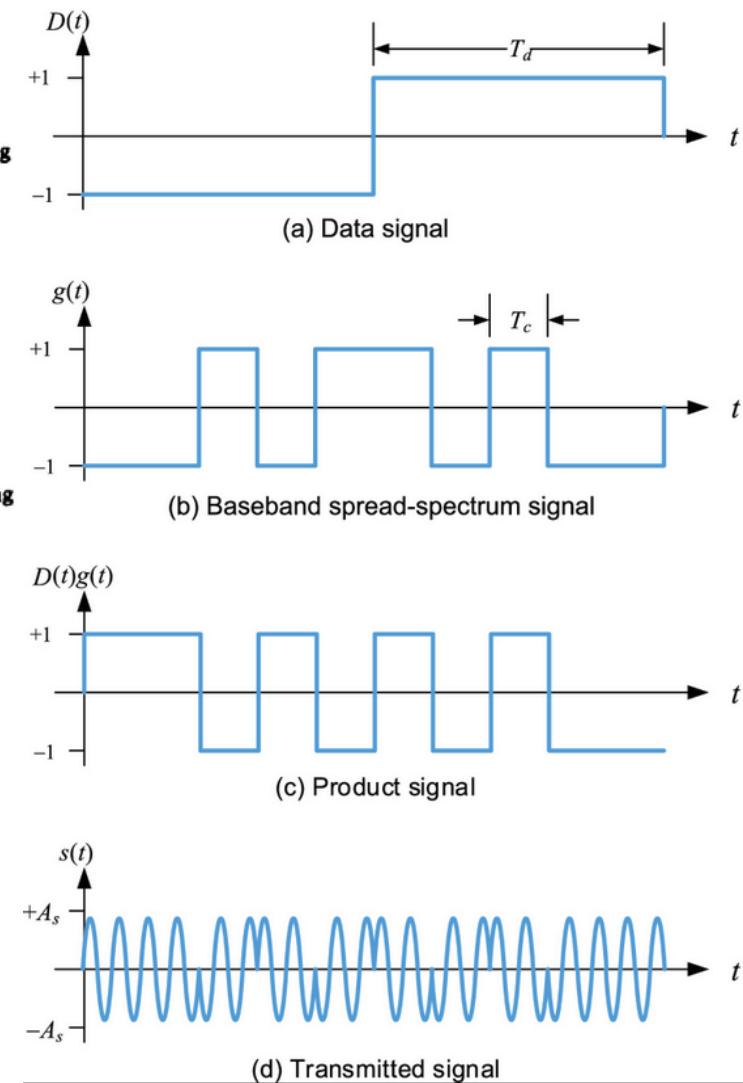
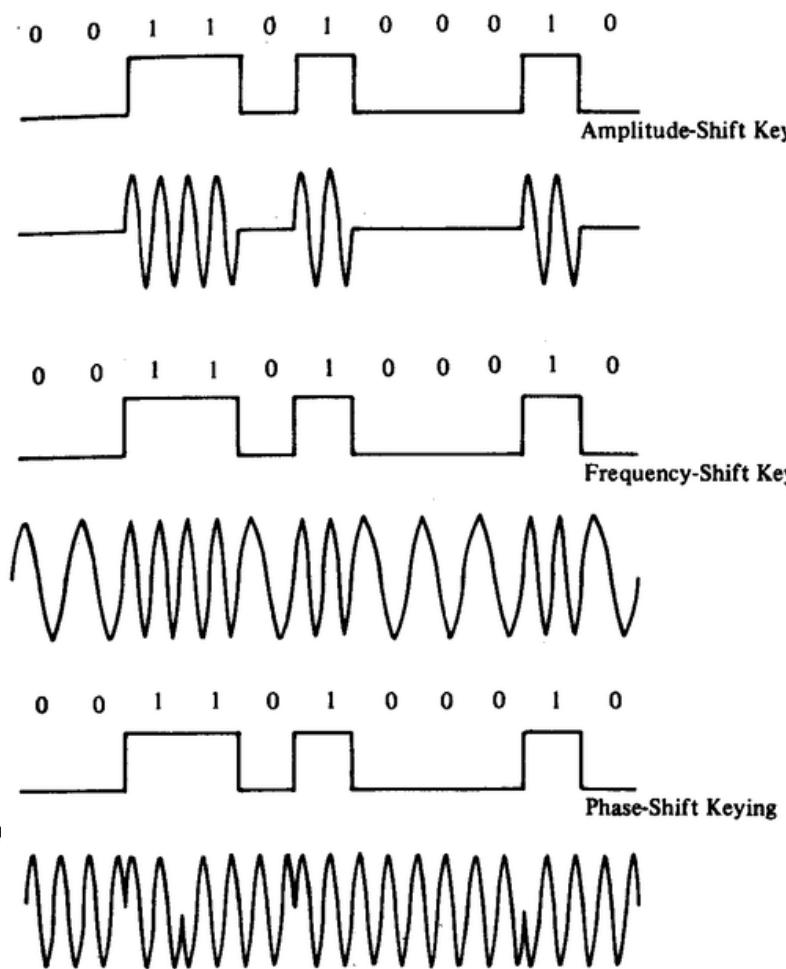
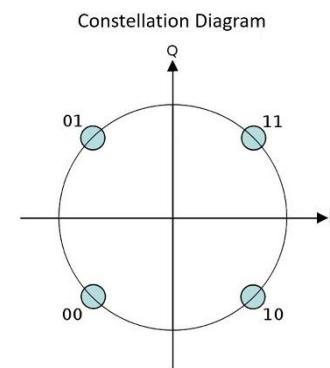
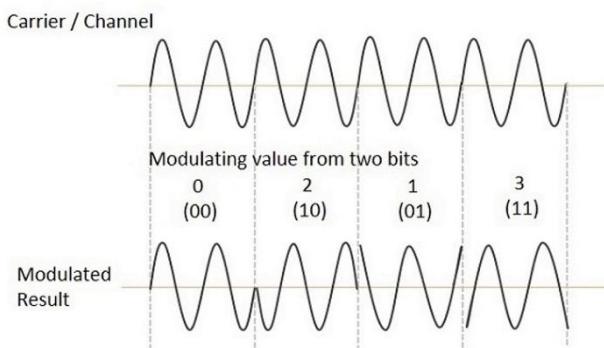
IEEE 802.15.4
2.4 GHz



IEEE 802.15.4

▶ Modulación:

- ▶ DSSS: Direct-Sequence Spread Spectrum
- ▶ PSK: Phase-Shift-Keying. Modulación por desplazamiento de fase
- ▶ QPSK: Quadrature-phase keying

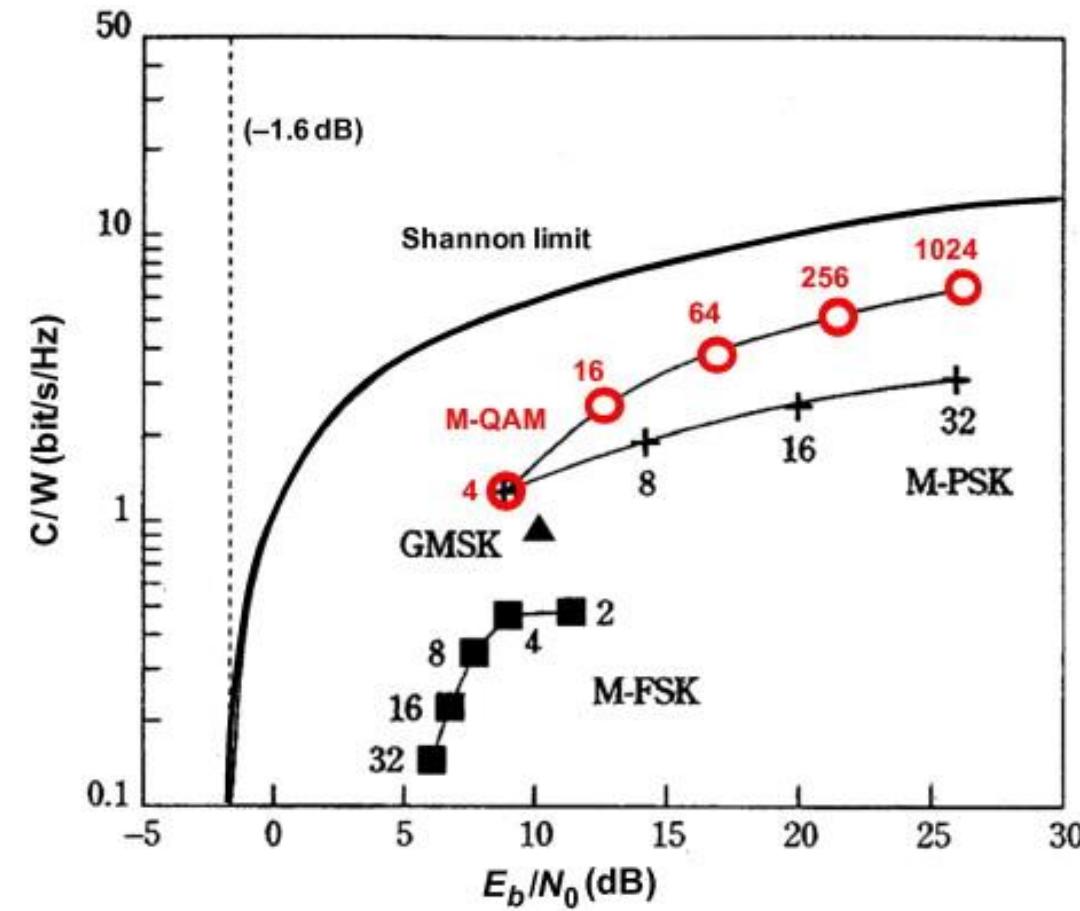


Inciso: Teorema de Shannon-Hartley

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

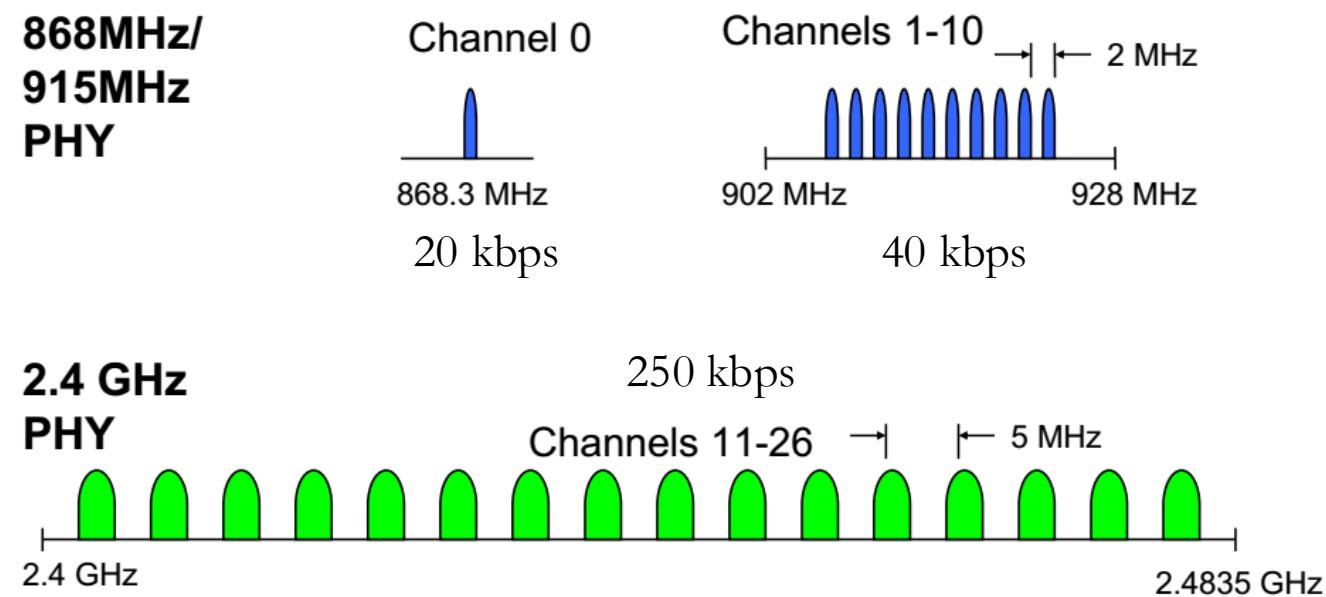
donde:

- B es el ancho de banda del canal en Hertzios.
- C es la capacidad del canal (tasa de bits de información bit/s)
- S es la potencia de la señal útil, que puede estar expresada en vatios, milivatios, etc., (W, mW, etc.)
- N es la potencia del ruido presente en el canal, (mW, μ W, etc.) que trata de enmascarar a la señal útil.



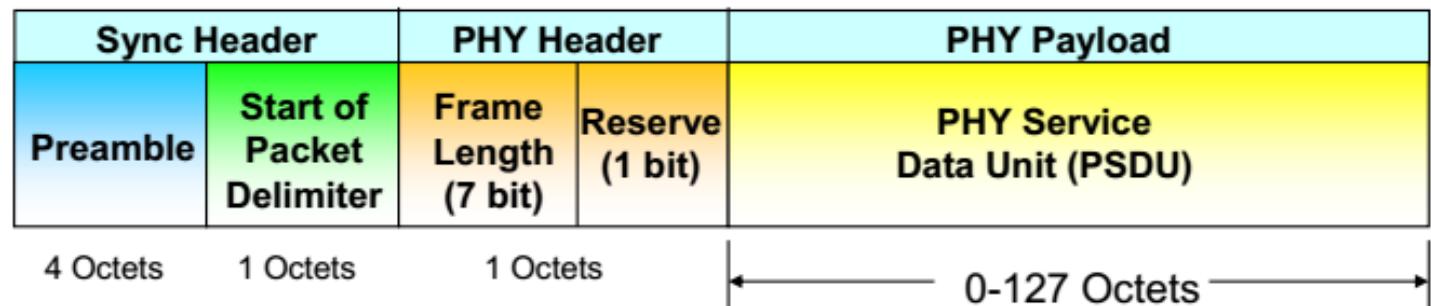
IEEE 802.15.4

- ▶ Nivel físico. Funciones:
 - ▶ Activación/desactivación del módulo radio
 - ▶ Detección de ocupación del canal para CSMA/CA
 - ▶ Indicador de calidad del enlace en la recepción de paquetes
 - ▶ Selección de canal de frecuencia (27 canales)
 - ▶ Transmisión y recepción de datos



IEEE 802.15.4

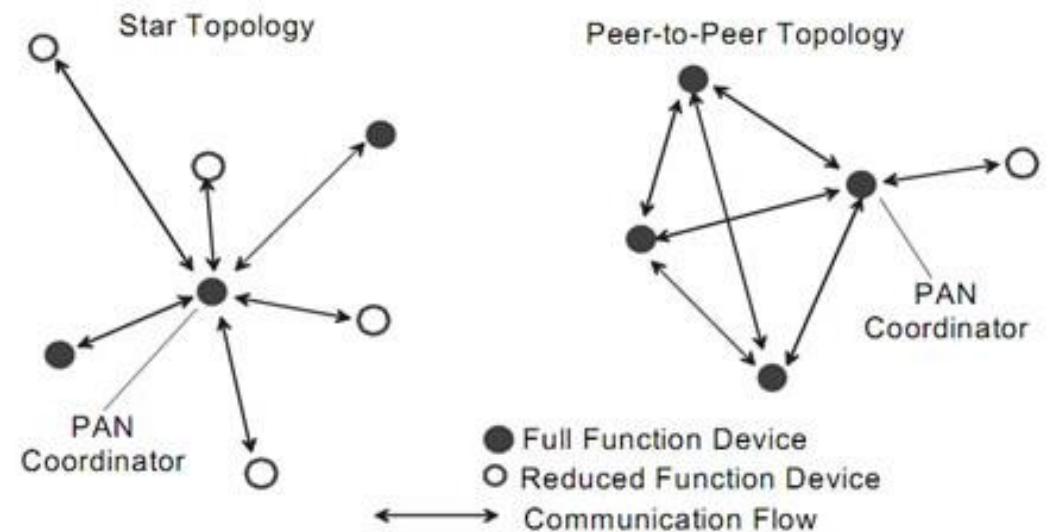
- ▶ Nivel físico. Formato de trama
 - ▶ Preámbulo: sincronización
 - ▶ Indicador de comienzo de paquete: “11100101”
 - ▶ Cabecera de nivel físico: tamaño de la trama
 - ▶ *Payload*: hasta 127 bytes



IEEE 802.15.4

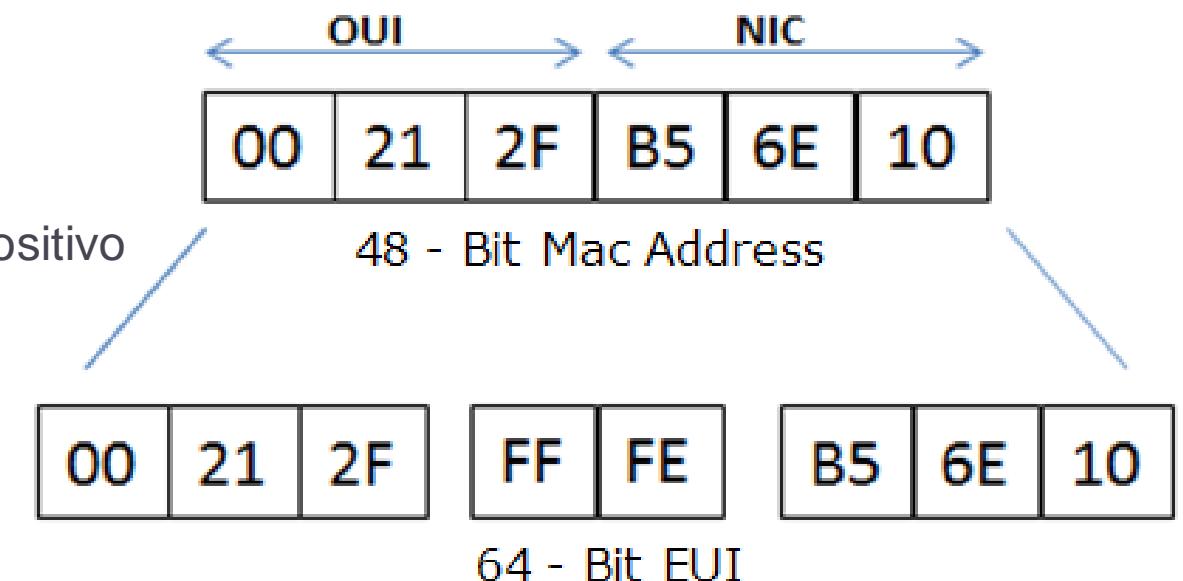
- ▶ Nivel de enlace
- ▶ Dispositivos físicos:
 - ▶ *Full-function device*, FFD
 - ▶ Puede ser coordinador de la PAN o nodo final
 - ▶ Tareas de enrutamiento
 - ▶ Se comunica con FFD y RFD
 - ▶ *Reduced-function device*, RFD
 - ▶ Implementación simple
 - ▶ Solo se comunican con FFD
 - ▶ No puede ser coordinador
- ▶ Dispositivos lógicos:
 - ▶ Dispositivo final
 - ▶ RFD o FFD sin tareas de control
 - ▶ Coordinador/enrutador
 - ▶ FFD con tareas de gestión y control de la red, básicamente, enrutar
 - ▶ Coordinador PAN
 - ▶ Controlador principal de la red. Una red solo puede tener uno. Gestiona direccionamiento

Topologías básicas

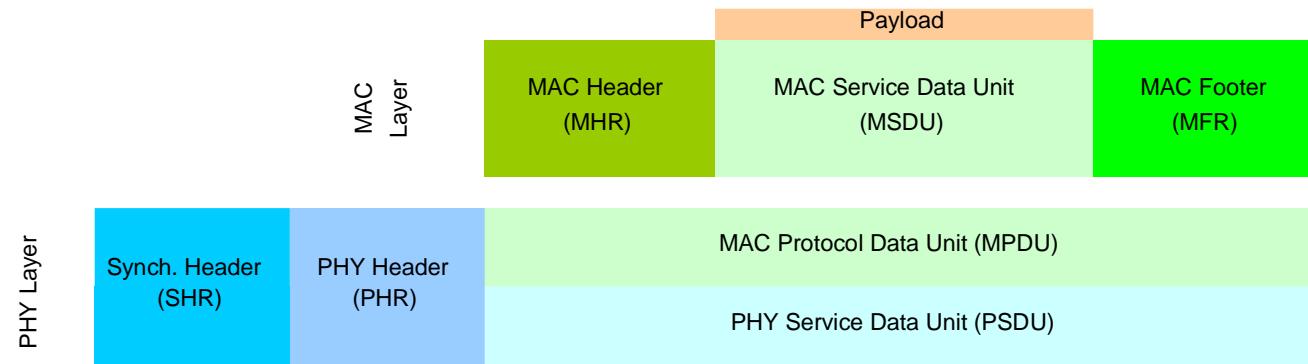


IEEE 802.15.4

- ▶ Nivel de enlace:
- ▶ Direccionamiento
 - ▶ 64 bits (obligatoria): MAC extendida (EUI-64). Identificador único universal
 - ▶ 16 bits (opcional): otorgada por el coordinador PAN cuando el dispositivo se asocia. Identificador sólo para la red
- ▶ Extended Unique Identifier (EUI-64)
 - ▶ 24 bits OUI (*Organizationally Unique Identifier*)
 - ▶ 16-bit 0xFFFF
 - ▶ 24 bits NIC (Network Interface Controller) del dispositivo



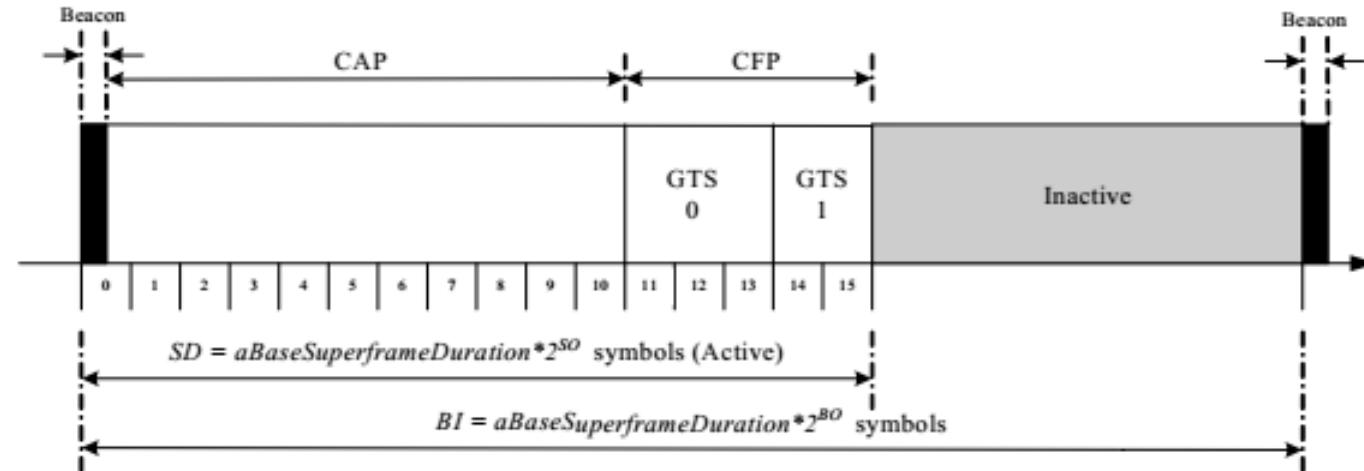
IEEE 802.15.4



- ▶ Nivel enlace
 - ▶ Algoritmo MAC basado en CSMA (Simple Carrier Sense Multiple Access)
 - ▶ Idea básica del algoritmo CSMA / CA:
 - ▶ Primero espera hasta que el canal esté inactivo.
 - ▶ Una vez que el canal esté libre, comience a enviar los data frames de datos después de un backoff aleatorio.
 - ▶ El receptor reconoce la correcta recepción de un data frame
 - ▶ Si el remitente no recibe un acuse de recibo, se reintenta la transmisión de datos.

IEEE 802.15.4

- ▶ Nivel de enlace:
- ▶ 2 modos: *beacon* y *beaconless*
 - ▶ *Beaconless*: acceso por contienda (CSMA/CA puro)
 - ▶ *Beacon*: acceso ranurado: concepto de “supertrama” (*superframe*)

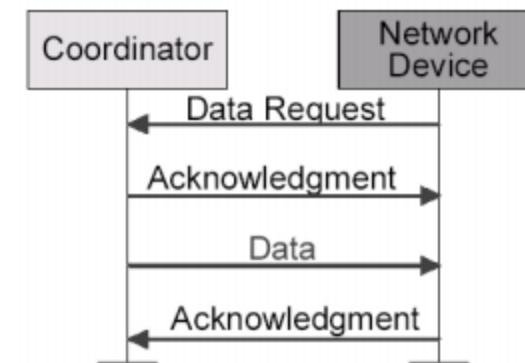
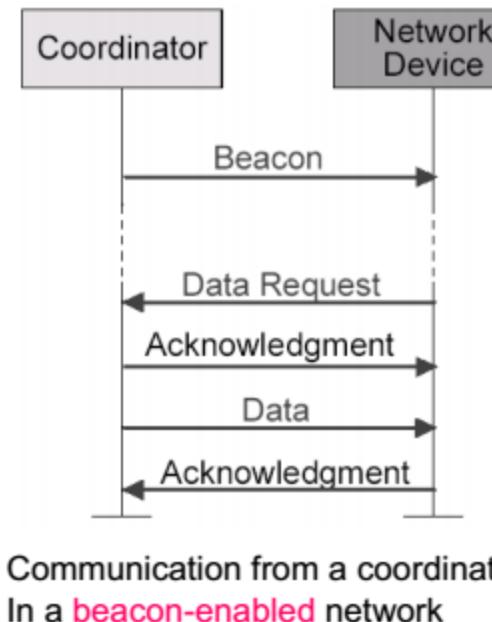


- ▶ CAP: *Contention Access Period*
- ▶ CFP: *Contention Free Period*. GTS: *Guaranteed Time Slots*
- ▶ *Inactive*: todos los dispositivos duermen

IEEE 802.15.4

- ▶ Nivel de enlace
- ▶ Transmisión de datos: reconocimiento de recepción (ACK)

Coordinador → Dispositivo final



Communication from a coordinator
in a **non beacon-enabled** network

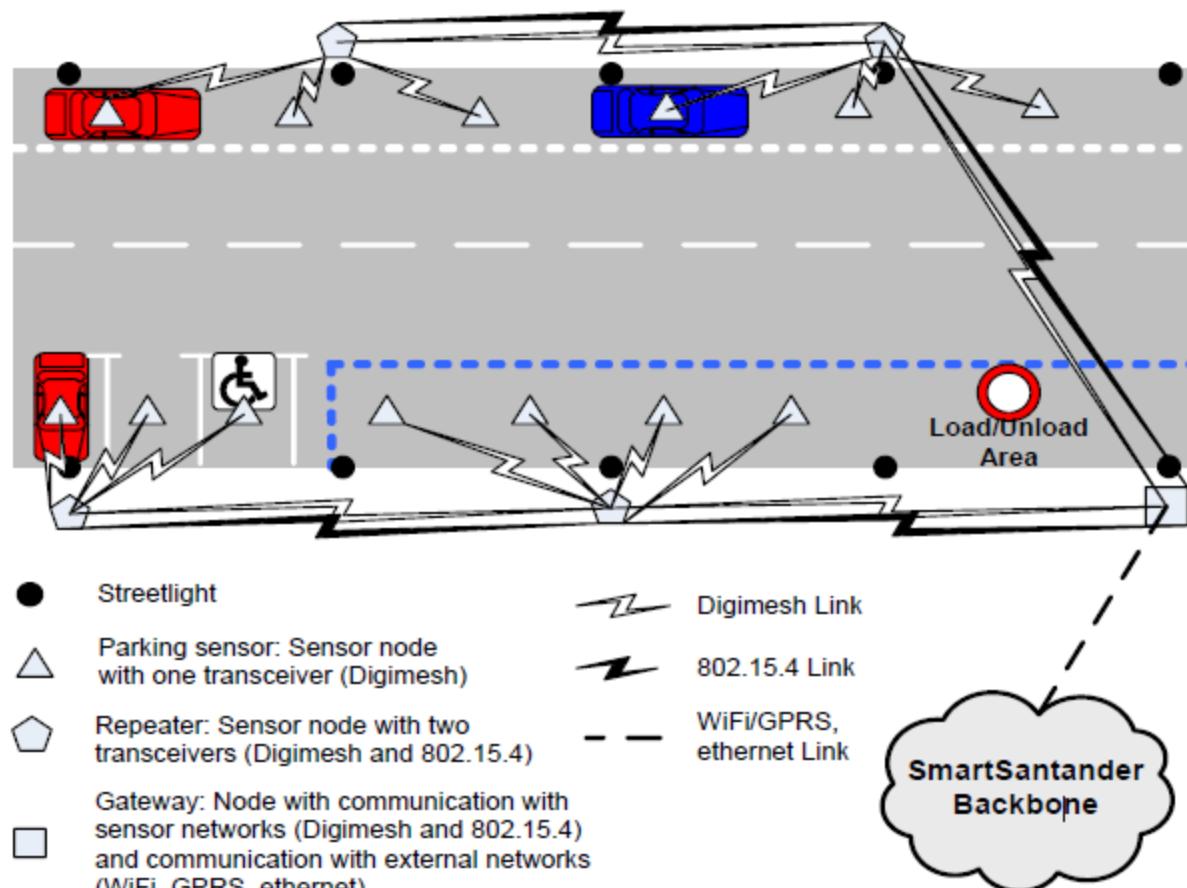
IEEE 802.15.4 – Caso de uso

- ▶ Smart-Santander: 3000 dispositivos 802.15.4



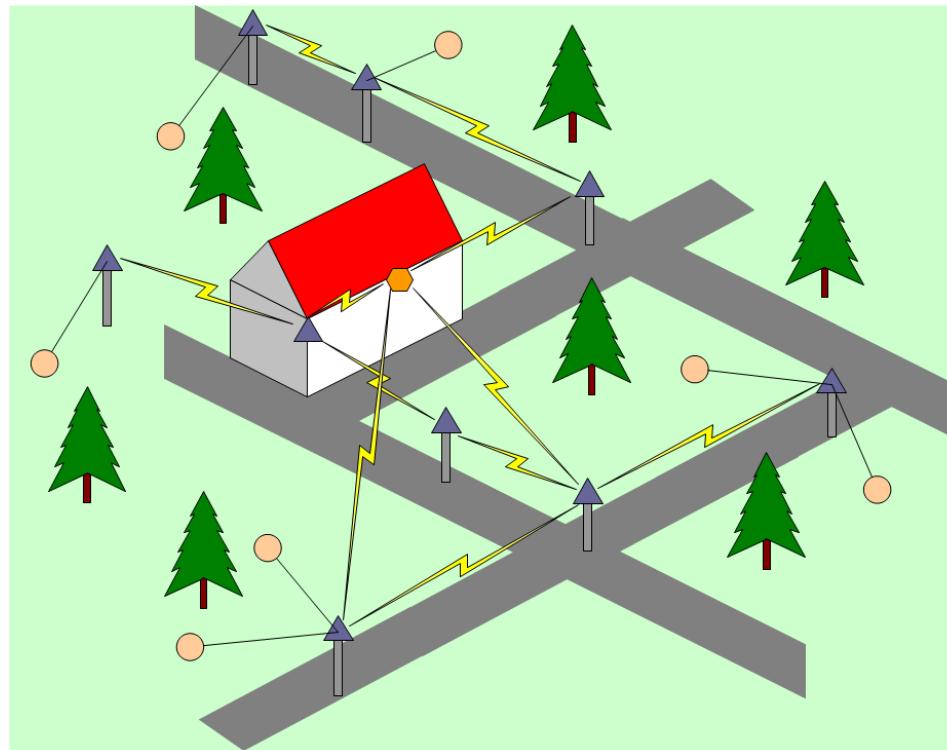
IEEE 802.15.4 – Caso de uso

- Smart-Santander: aplicaciones: control de aparcamiento



IEEE 802.15.4 – Caso de uso

- Smart-Santander: aplicaciones: control de riego



● Park irrigation monitoring sensor. To be deployed buried in the ground.

▲ Repeater. To be deployed at available street lights or traffic lights.

◆ Gateway. Connected to Internet/Intranet.

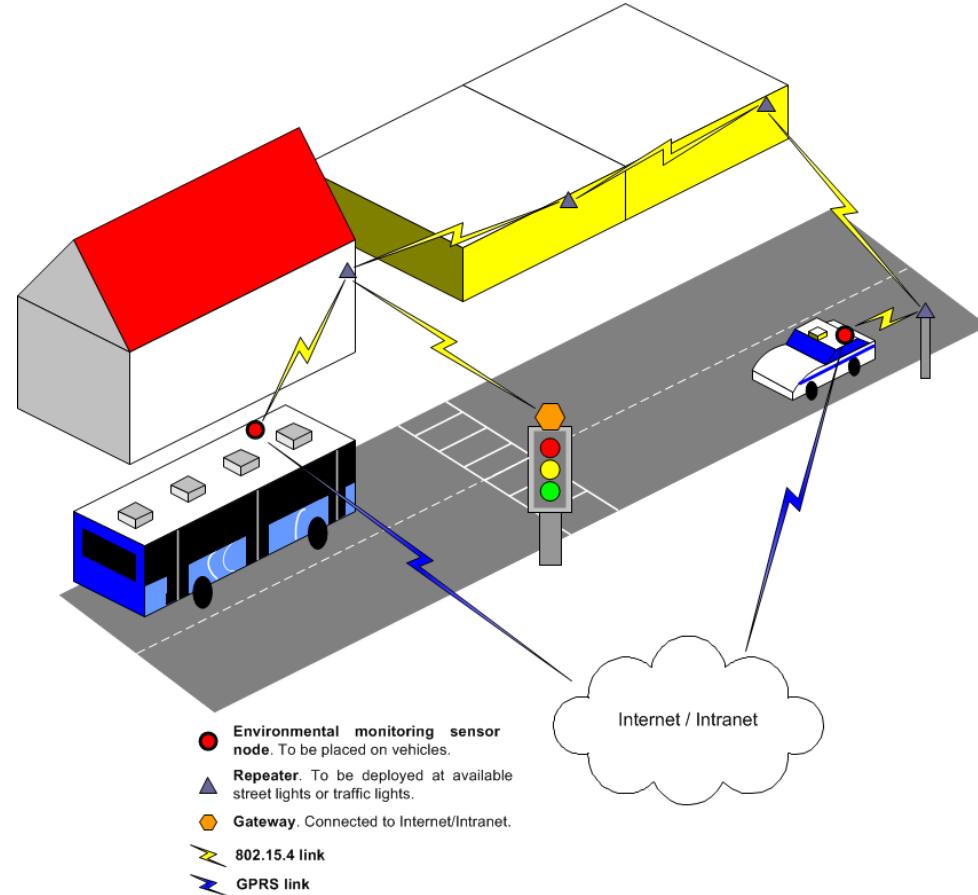
Radio link

Wired link

IEEE 802.15.4 – Caso de uso

- Smart-Santander: aplicaciones: control de transportes y polución de tráfico

<http://maps.smartsantander.eu/>



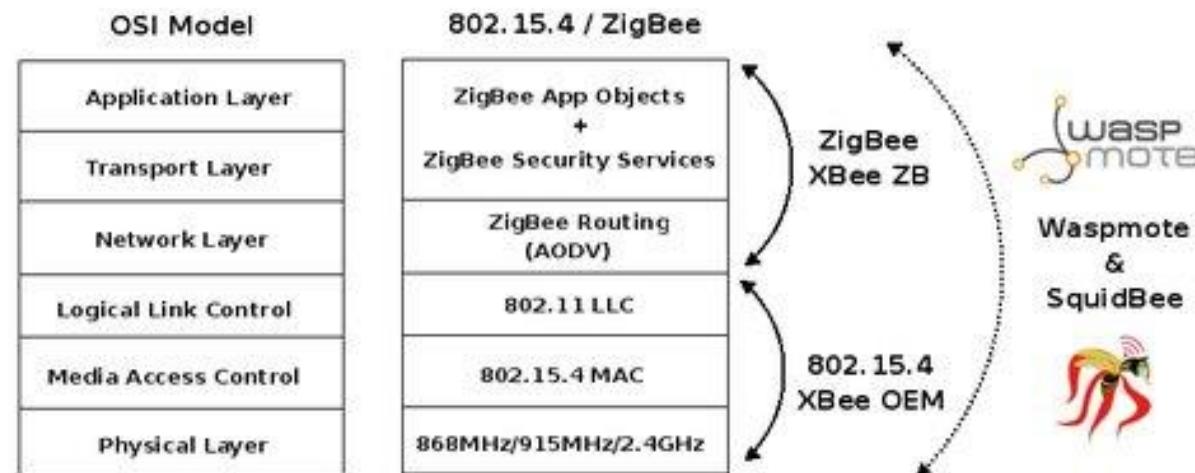
Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

Zigbee



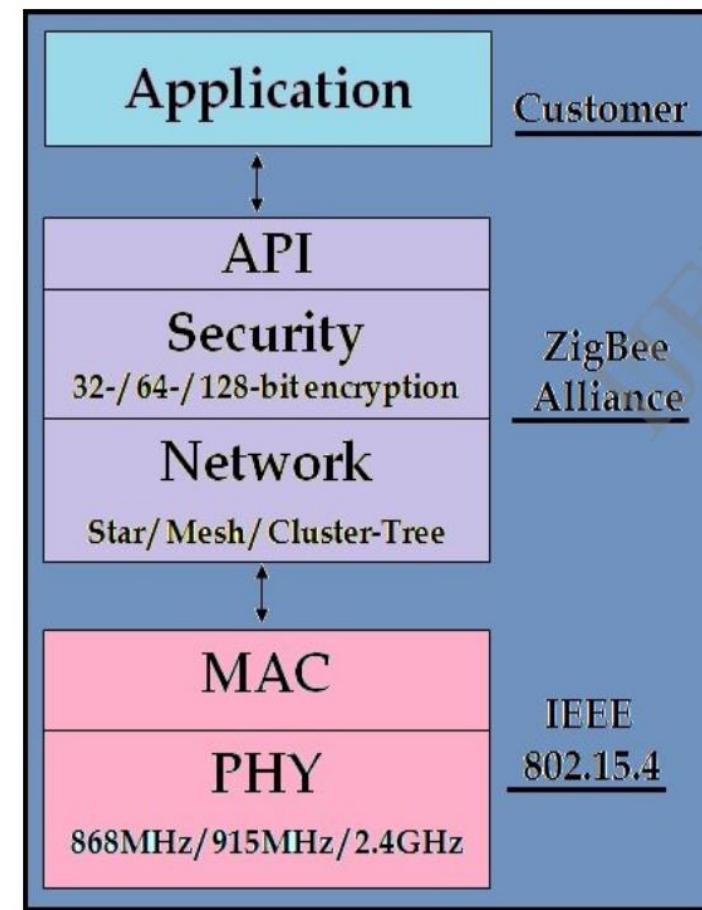
- ▶ Protocolo estándar abierto que provee funcionalidad y características adicionales sobre 802.15.4
- ▶ Promovido por la *Zigbee Alliance* desde 2002.



Zigbee

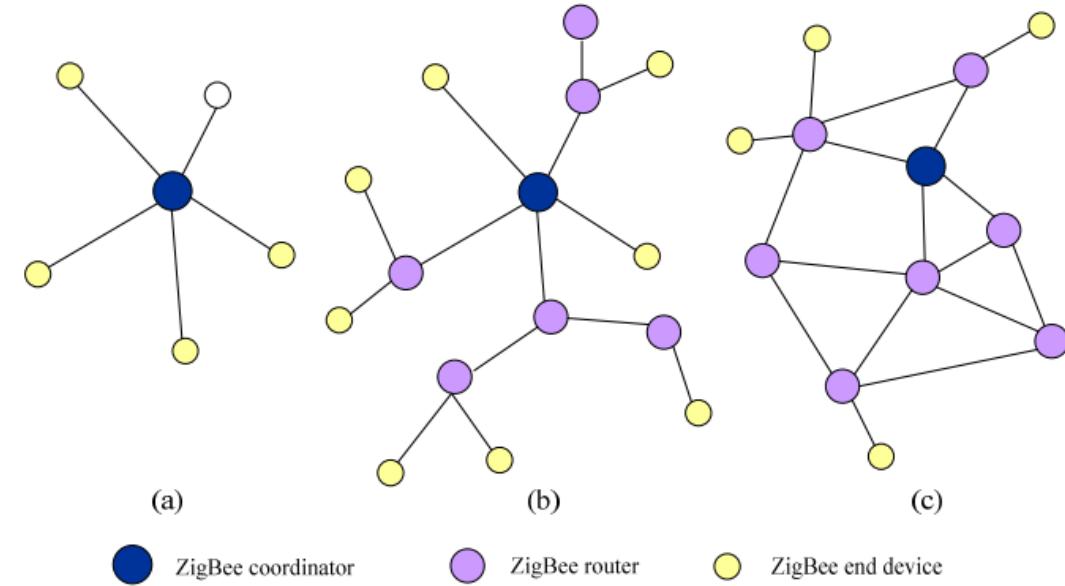
▶ Características adicionales a IEEE 802.15.4

- ▶ Servicios añadidos de cifrado: a nivel de red y aplicación
- ▶ Asociación y autenticación: durante el proceso de asociación a la red
- ▶ Direccionamiento a nivel de red
- ▶ Enrutamiento: basado en árbol o protocolo reactivo basado en AODV



Zigbee

- ▶ Topologías
- ▶ Topología en estrella
 - ▶ Comunicación a través del coordinador PAN
 - ▶ Hojas pueden ser una combinación de FFD y RFD
 - ▶ Coordinador PAN suele ser un dispositivo confiable conectado a la red eléctrica
- ▶ Topología *Peer-to-peer (mesh)*
 - ▶ Extensión de la topología en estrella para comunicación directa entre dispositivos
 - ▶ Enrutamiento
- ▶ Topología en árbol clusterizada
 - ▶ Varios coordinadores conectados entre sí dan servicio a nodos finales
 - ▶ Un coordinador es designado coordinador PAN

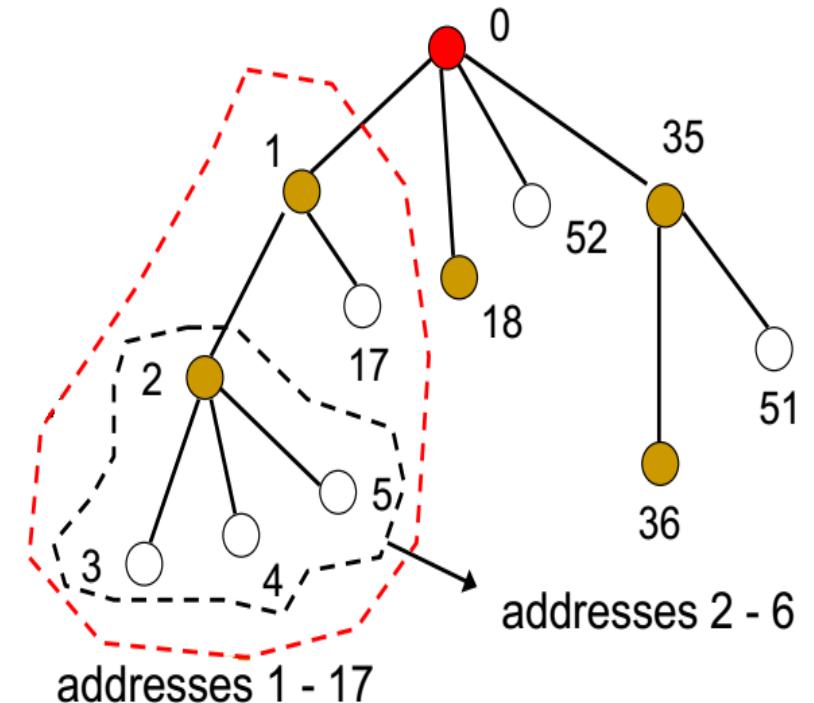


Tipos de nodos lógicos

- ▶ Dispositivo final
 - ▶ RFD o FFD sin tareas de control
- ▶ Router Zigbee
 - ▶ FFD con tareas de gestión y control de la red
- ▶ Coordinador Zigbee
 - ▶ Controlador principal de la red. Una red solo puede tener uno

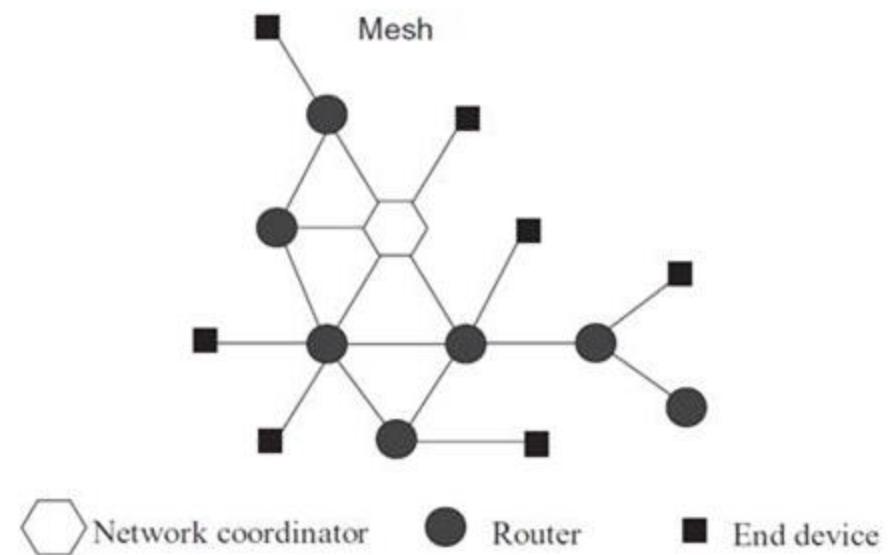
Zigbee

- ▶ Direccionamiento (i): *Distributed Address Assignment Mechanism (DAAM)*
- ▶ Direccionamiento jerárquico
- ▶ Se reservan unas direcciones para cada rama
- ▶ Los nodos en la misma rama tienen direcciones contiguas
- ▶ La dirección de cada nodo es asignada por el padre (*self-organizing*)
- ▶ Permite simplificar el enrutamiento (basado en árbol): no hay que hacer búsqueda de direcciones una a una
- ▶ Complejidad de escalabilidad



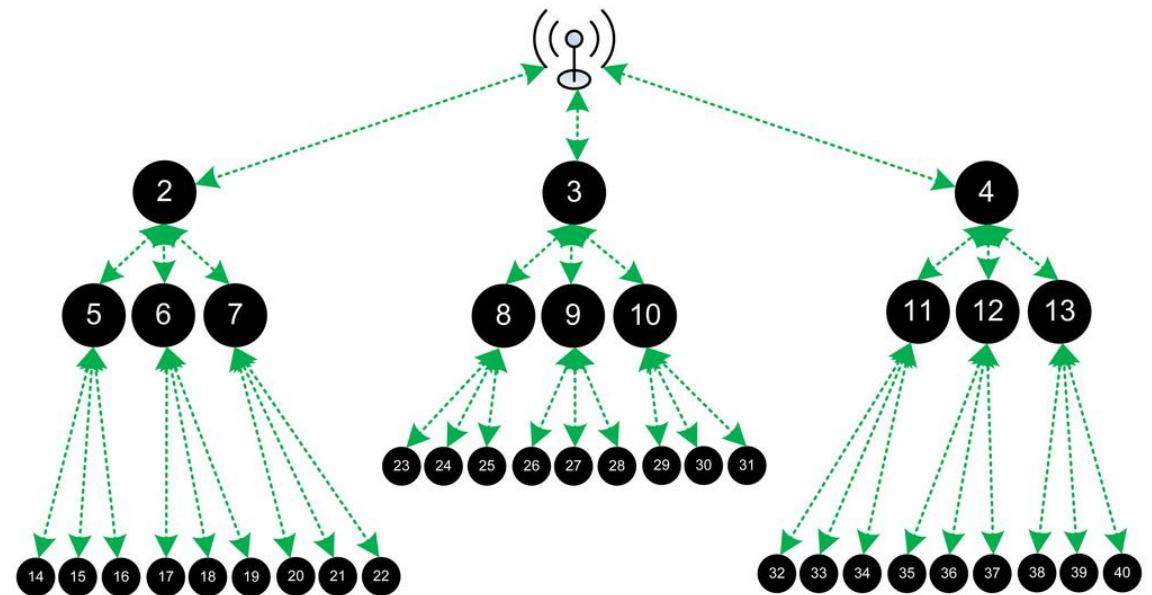
Zigbee

- ▶ Direccionamiento (ii): *Stochastic Address Assignment Mechanism* (SAAM)
- ▶ Direccionamiento no jerárquico
- ▶ Los dispositivos calculan su dirección de forma aleatoria e independiente
- ▶ Enrutamiento *mesh* (AODV)
- ▶ Pueden haber conflictos de direcciones duplicadas
- ▶ Los nodos requieren de *broadcasts* frecuentes para mantener las rutas



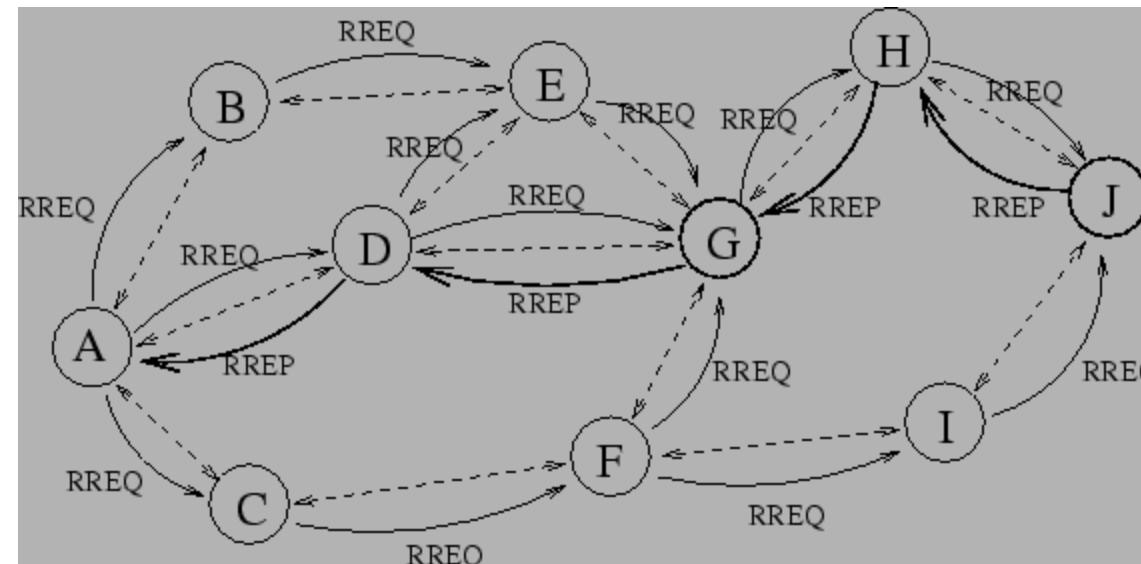
Zigbee

- ▶ Enrutamiento: basado en árbol jerárquico
- ▶ Cuando un *router* recibe un paquete, comprueba si la dirección destino está en su rama
 - ▶ Si es así, el *router* acepta el paquete y lo reenvía a su destino
 - ▶ Si no es así, el *router* reenvía el paquete al siguiente *router*
- ▶ Se evitan tablas de enrutamiento: grupos de direcciones
- ▶ Muy poco *overhead* en la red



Zigbee

- ▶ Enrutamiento: basado en redes *mesh*
- ▶ Protocolo enrutamiento similar a AODV (*Ad hoc On-Demand Distance Vector Routing*)
 - ▶ Protocolo reactivo: descubrimiento de ruta cada vez que se quiera enviar un paquete
 - ▶ Métrica: PDR
- ▶ Mayor necesidad de memoria y procesado por los nodos



Zigbee

	Pros	Cons
Star	<ol style="list-style-type: none">1. Easy to synchronize2. Support low power operation3. Low latency	<ol style="list-style-type: none">1. Small scale2. Single Point of Failure
Tree	<ol style="list-style-type: none">1. Low routing cost2. Can form superframes to support sleep mode3. Allow multihop communication	<ol style="list-style-type: none">1. Route reconstruction is costly2. Latency may be quite long
Mesh	<ol style="list-style-type: none">1. Robust multihop communication2. Network is more flexible3. Lower latency	<ol style="list-style-type: none">1. Cannot form superframes (and thus cannot support sleep mode)2. Route discovery is costly3. Needs storage for routing table

Zigbee – Caso de uso

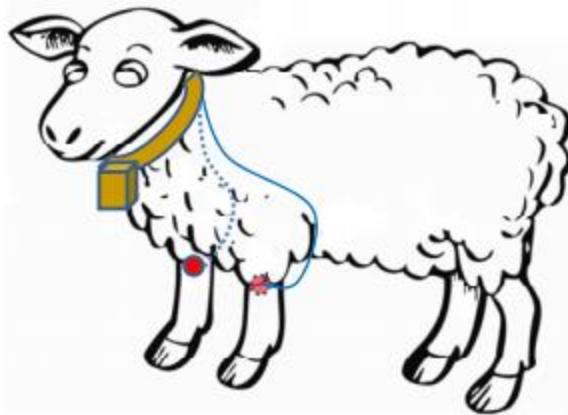
- ▶ ¡Qué viene el lobo!



F. Llario, S. Sendra, L. Parra and J. Lloret, "Detection and protection of the attacks to the sheep and goats using an intelligent wireless sensor network," *IEEE International Conference on Communications Workshops (ICC)*, Budapest, 2013, pp. 1015-1019. doi: 10.1109/ICCW.2013.6649385

Zigbee – Caso de uso

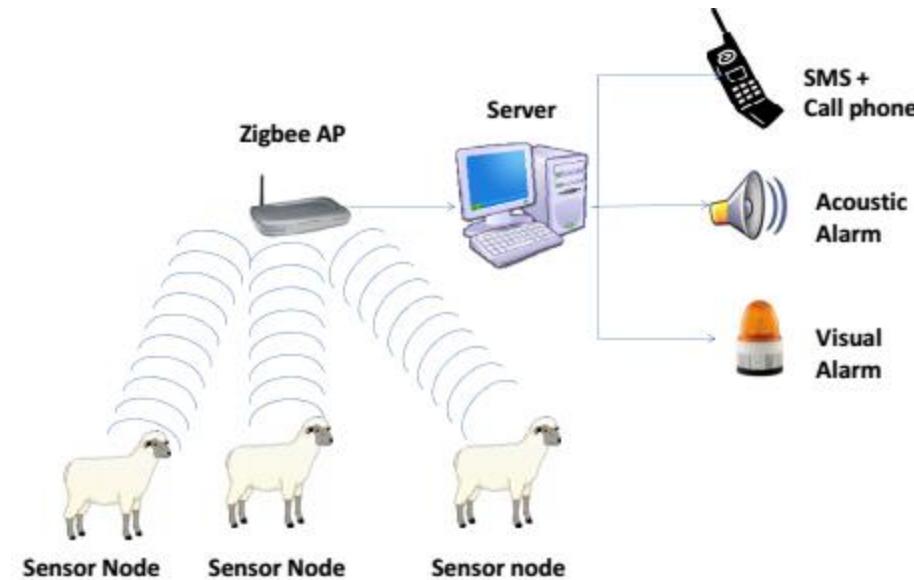
- ▶ Sensores de constantes vitales en cada oveja



F. Llario, S. Sendra, L. Parra and J. Lloret, "Detection and protection of the attacks to the sheep and goats using an intelligent wireless sensor network," *IEEE International Conference on Communications Workshops (ICC)*, Budapest, 2013, pp. 1015-1019. doi: 10.1109/ICCW.2013.6649385

Zigbee – Caso de uso

▶ Sistema de alarma



- ▶ Protocolo de alarma: se dispara la alerta cuando se reciben 5 alarmas de 5 ovejas distintas durante un periodo de 5 minutos

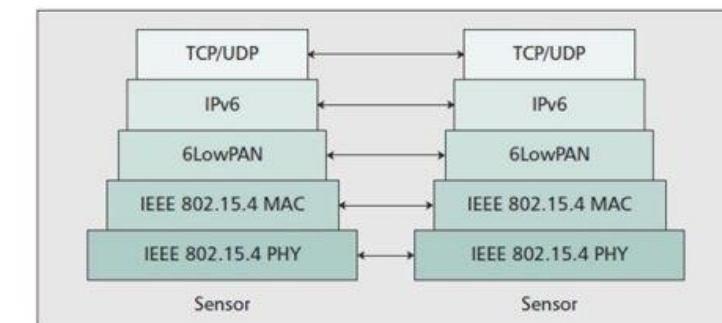
Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ **6LoWPAN – IPv6 over low power Wireless Personal Area Networks**
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

6LoWPAN - Introducción



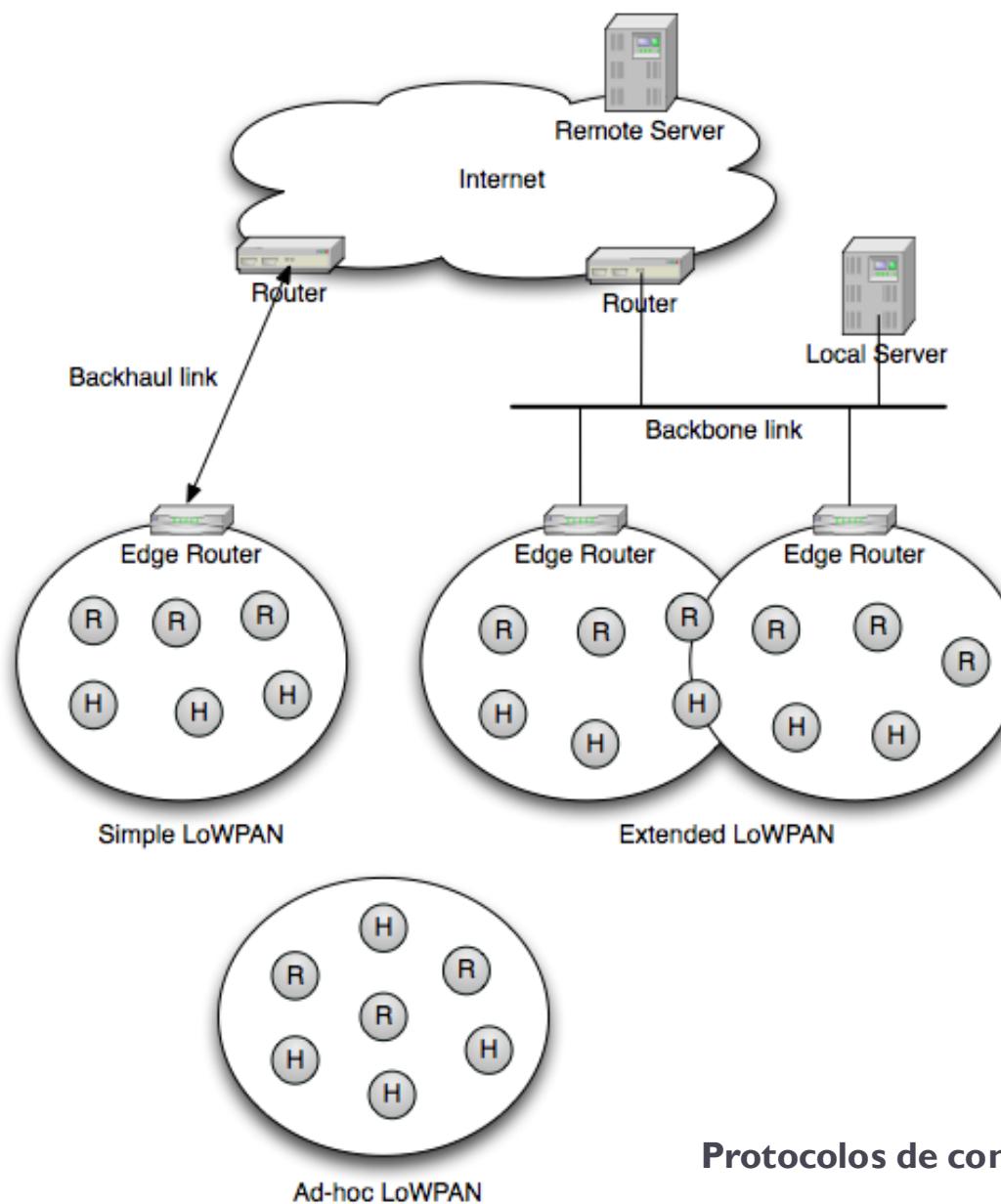
- ▶ Capa de adaptación para transportar paquetes IPv6 sobre Low-Power Wireless Personal Area Networks (LP-WPAN)
 - ▶ Definido sobre el estándar IEEE 802.15.4
 - ▶ Está siendo adaptado también para otros protocolos a nivel de enlace (Bluetooth Smart, Low-power Wi-Fi, Power Line Control (PLC))
- ▶ Interoperabilidad con otras tecnologías
- ▶ Integración con **Internet** transparente
 - ▶ Permite el uso de API de sockets estándar
- ▶ Uso mínimo de código y memoria
- ▶ Escalabilidad global
- ▶ Flujos de datos **end-to-end**
- ▶ Soporte fragmentación
- ▶ Soporte de routing IP (ej. IETF RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) RFC 5560)
- ▶ Soporte unicast, multicast y broadcast



6LoWPAN - Introducción

- ▶ Problemas de IPv6 para trabajar en WSN:
 - ▶ Direcciones IPv6 muy largas (128 bits): soporte para direcciones de 64 bits y 16 bits 802.15.4
 - ▶ Cabecera IPv6 muy larga: compresión eficiente de cabeceras
 - ▶ Compresión IPv6
 - ▶ Compresión cabeceras de extensión IPv6
 - ▶ Compresión cabecera UDP
 - ▶ Autoconfiguración de red → uso de *network Discovery* con intensos envíos multicast
 - ▶ Los protocolos de enrutamiento ad-hoc suelen introducir mucho overhead → 6LoWPAN emplea su propio protocolo de enrutamiento eficiente (RPL), aunque es agnóstico al protocolo de enrutamiento
- ▶ Estándares del IETF sobre 6LoWPAN:
 - ▶ RFC 4944 - cabeceras
 - ▶ RFC 6282 - formato de compresión
 - ▶ RFC 6550 - *routing* (RPL)
 - ▶ RFC 6775 - *neighbour discovery*

6LoWPAN - Arquitectura



▶ Edge router:

- ▶ Intercambio de datos entre dispositivos 6LoWPAN e Internet (otras redes IPv6)
- ▶ Intercambio de datos entre dispositivos 6LoWPAN
- ▶ Opcional: soporte para conectar redes 6LoWPAN a redes IPv4

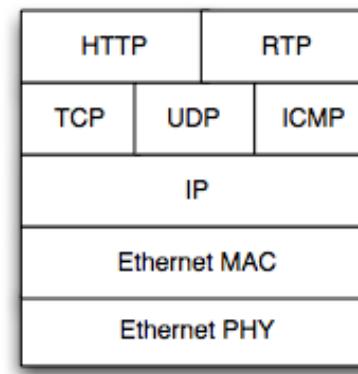
IPv6	
Ethernet MAC	LoWPAN Adaptation
	IEEE 802.15.4 MAC
Ethernet PHY	IEEE 802.15.4 PHY

IPv6-LoWPAN Router Stack

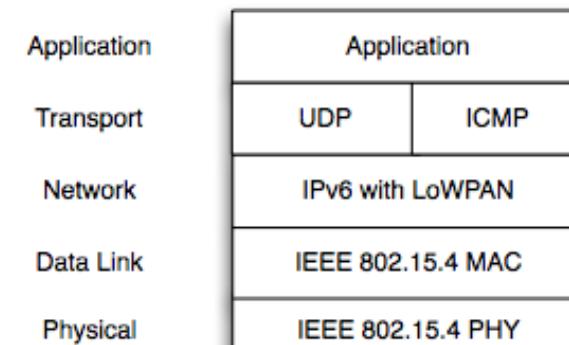
6LoWPAN - Características

- ▶ 6LoWPAN hace uso de compresión de las direcciones IPv6: 128 b → 64 o 16 b
- ▶ RFC 4944:
 - ▶ Introduce una cabecera básica
 - ▶ Compresión de formatos para HC1 (IPv6 header) and HC2 (UDP header)
 - ▶ Fragmentación/ reensamblado
 - ▶ Soporte cabecera para la red *mesh*
 - ▶ Mapeo multicast a espacio de direcciones de 16-bit
- ▶ RFC 6282:
 - ▶ Nuevos formatos compresión IPHC (IPv6 header) y NHC (Next-header)
 - ▶ Soporte para compresión global de direcciones (con contextos)
 - ▶ Soporte para compresión de la cabecera de opciones IPv6

TCP/IP Protocol Stack



6LoWPAN Protocol Stack

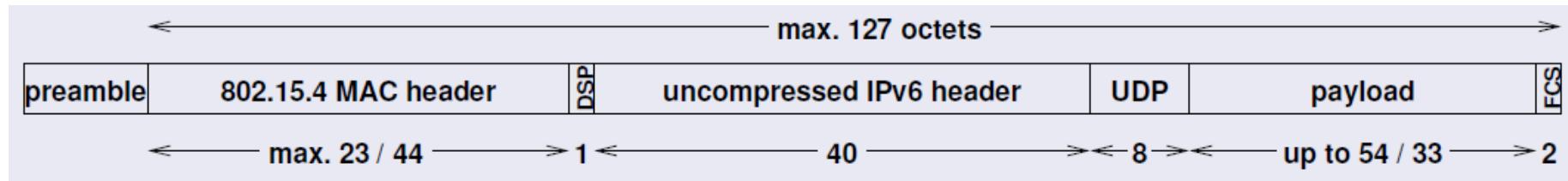


6LoWPAN - Direcccionamiento

- ▶ Direcciones IPv6 se comprimen en 6LoWPAN
- ▶ 6LoWPAN funciona en principio:
 - ▶ Direccionamiento plano (sin prefijos)
 - ▶ Direcciones de 64-bit or 16-bit
- ▶ 6LoWPAN comprime las direcciones IPv6
 - ▶ Quitando el prefijo de IPv6 (primeros 64 bits)
 - ▶ El prefijo global ya es conocido por todos los nodos de la red
 - ▶ Comprimiendo el identificador de la interfaz (IID)
 - ▶ Se quita para comunicaciones locales
 - Asume las direcciones de la MAC de nivel de enlace para el nivel 3
 - Requiere que las IPv6 se construyan a partir de la MAC
 - ▶ Se comprime en direcciones mutihop dst/src
 - ▶ Compresión utilizando un “contexto” bien conocido (RFC-6282)
 - ▶ Compresión de direcciones multicast

6LoWPAN - Formato de cabecera

- ▶ Peor escenario: Sin compresión

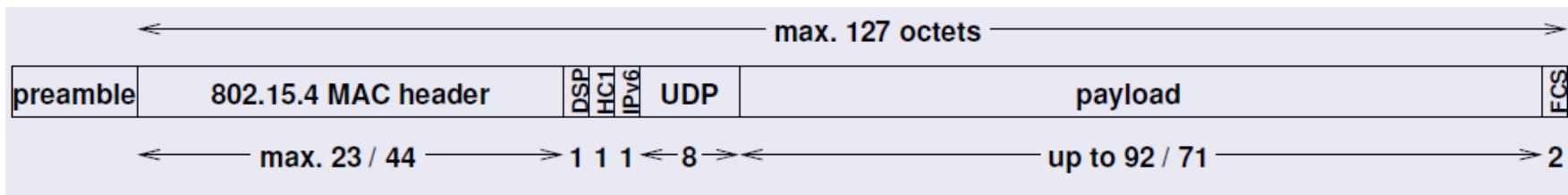


- ▶ Códigos Dispatch DSP (primer Byte)
 - ▶ Indica tipo de cabecera que viene a continuación

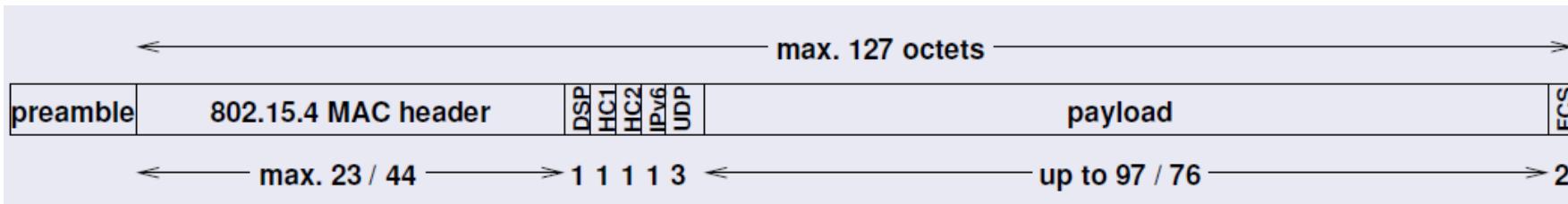
Bit Pattern	Short Code	Description
00 xxxxxx	NALP	Not A LoWPAN Packet
01 000001	IPv6	uncompressed IPv6 addresses
01 000010	LOWPAN_HC1	HC1 Compressed IPv6 header
01 010000	LOWPAN_BC0	BC0 Broadcast header
01 111111	ESC	Additional Dispatch octet follows
10 xxxxxx	MESH	Mesh routing header
11 000xxx	FRAG1	Fragmentation header (first)
11 100xxx	FRAGN	Fragmentation header (subsequent)

6LoWPAN – Compresión de cabecera

- ▶ Compresión cabecera IPv6



- ▶ Mejor caso, compresión cabecera IPv6 y UDP



6LoWPAN - Fragmentación

- ▶ Cálculo de la cabecera.
 - ▶ Cabecera IPv6 → 40 octetos
 - ▶ Cabecera UDP → 8 octetos
 - ▶ Cabecera MAC de 802.15.4
 - ▶ 25 octetos (sin seguridad)
 - ▶ $25+21=46$ octetos (si se utiliza seguridad AES-CCM-128)
 - ▶ Tamaño trama 802.15.4 es 127 octetos, queda para datos:
 - ▶ $127-25-40-8 = \textbf{54 octetos}$ (sin seguridad)
 - ▶ $127-46-40-8 = \textbf{33 octetos}$ (usando AES-CCM-128)
 - ▶ Tamaño máximo (MTU) paquete IPv6 es 1280 octetos
- 
- ▶ Se necesita fragmentación y reensamblado para encajar los 1280 octetos IPv6 en los 33 (o 54) octetos de 802.15.4

6LoWPAN - Fragmentación

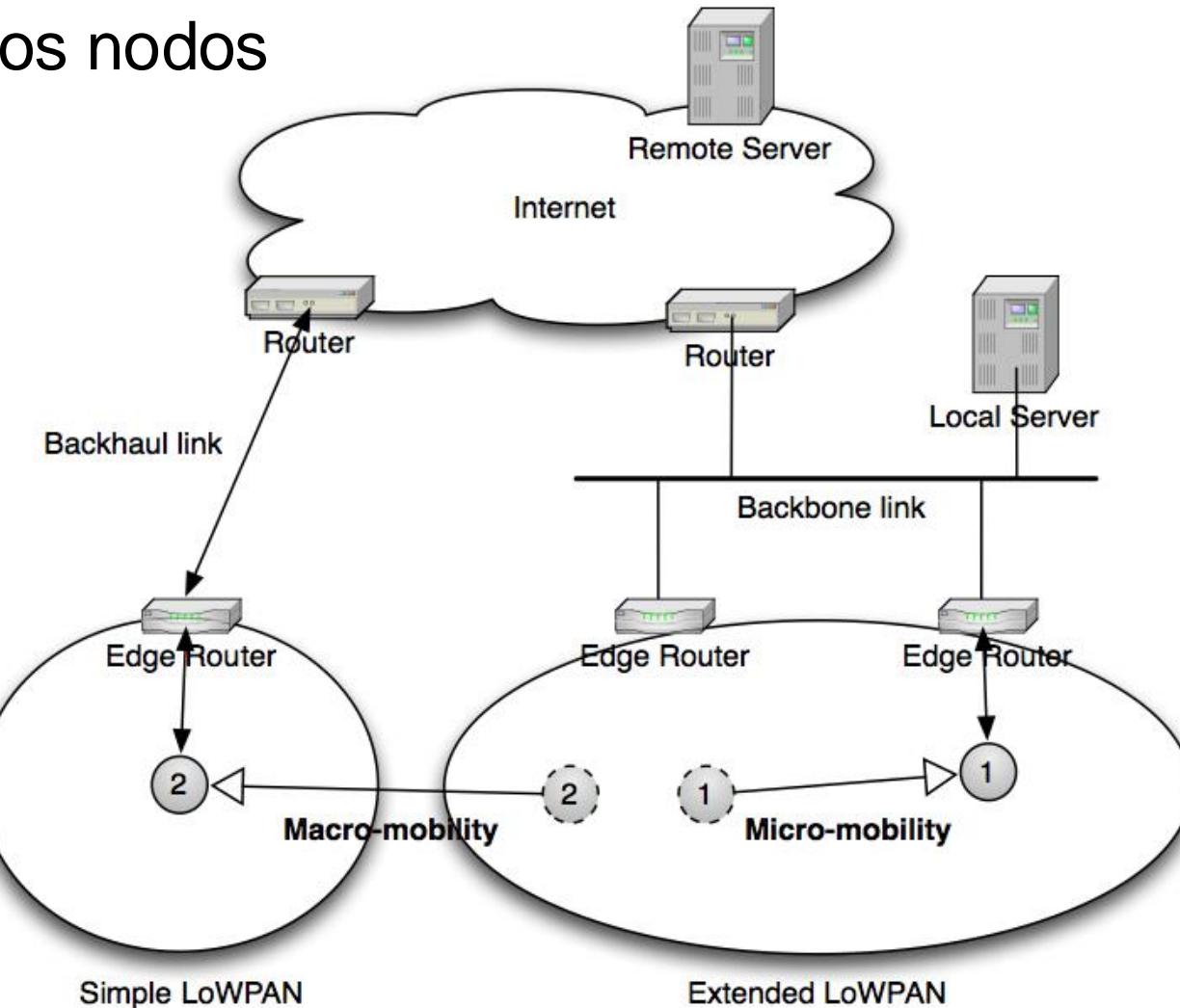
- ▶ 6LoWPAN define fragmentación y re-ensamblado IPv6
- ▶ Bajo rendimiento de paquetes grandes IPv6 fragmentados sobre redes LoWPAN
 - ▶ La pérdida de fragmentos causan que los paquetes se retransmitan
 - ▶ Bajo ancho de banda y alto retardo del canal inalámbrico
 - ▶ Los protocolos de aplicación sobre 6LoWPAN deben evitar fragmentación
 - ▶ Los protocolos de aplicación deberían aplicar compresión cuando se usan en 6LoWPAN

6LoWPAN - Autoconfiguración

- ▶ Autoconfiguración es importante para redes de 6LoWPAN con sistemas embebidos
- ▶ Inicialización de la red 6LoWPAN
 - ▶ 1. Mismos parámetros de conectividad a nivel enlace 802.15.4 entre nodos (*commissioning*)
 - ▶ 2. Configuración de direcciones de red, descubrimiento de vecinos, registros (*bootstrapping*)
 - ▶ 3. Establecimiento de rutas (*route initialization*)
 - ▶ 4. Mantenimiento continuo de los puntos anteriores

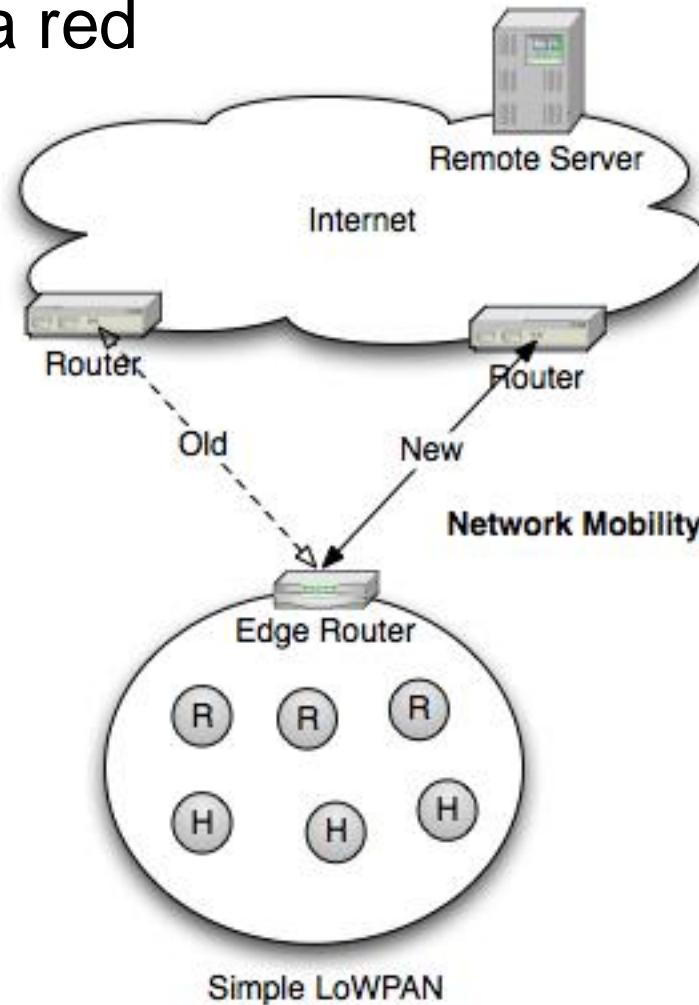
6LoWPAN - Movilidad

► Movilidad de los nodos



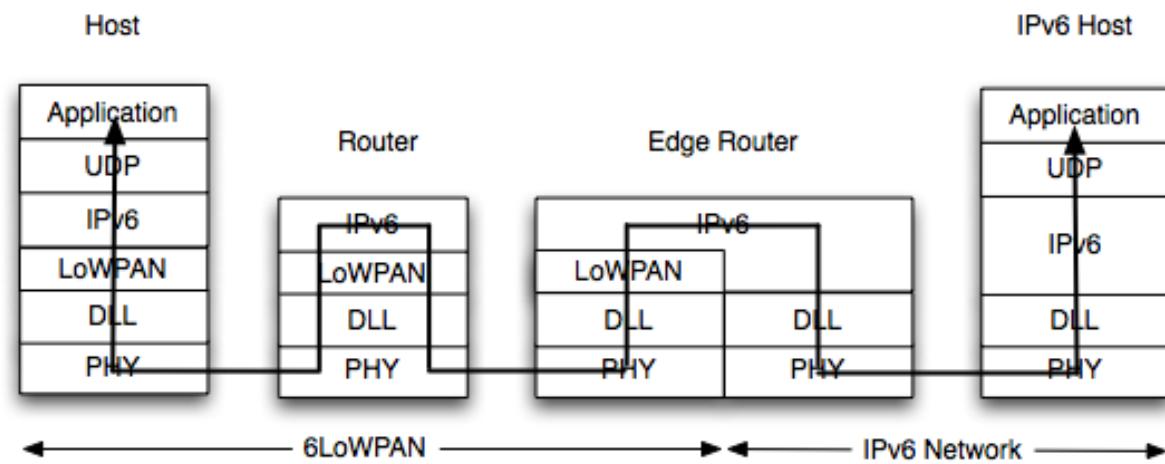
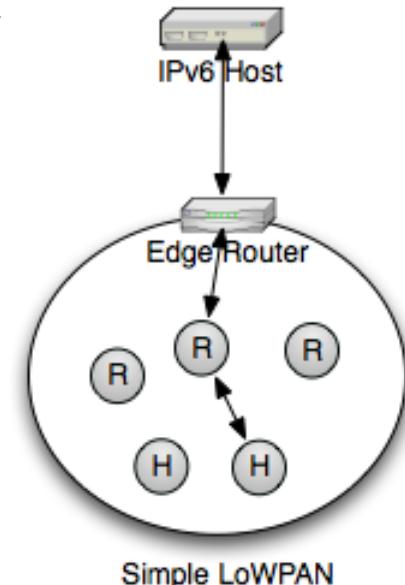
6LoWPAN - Movilidad

► Movilidad de la red



6LoWPAN - Encaminamiento

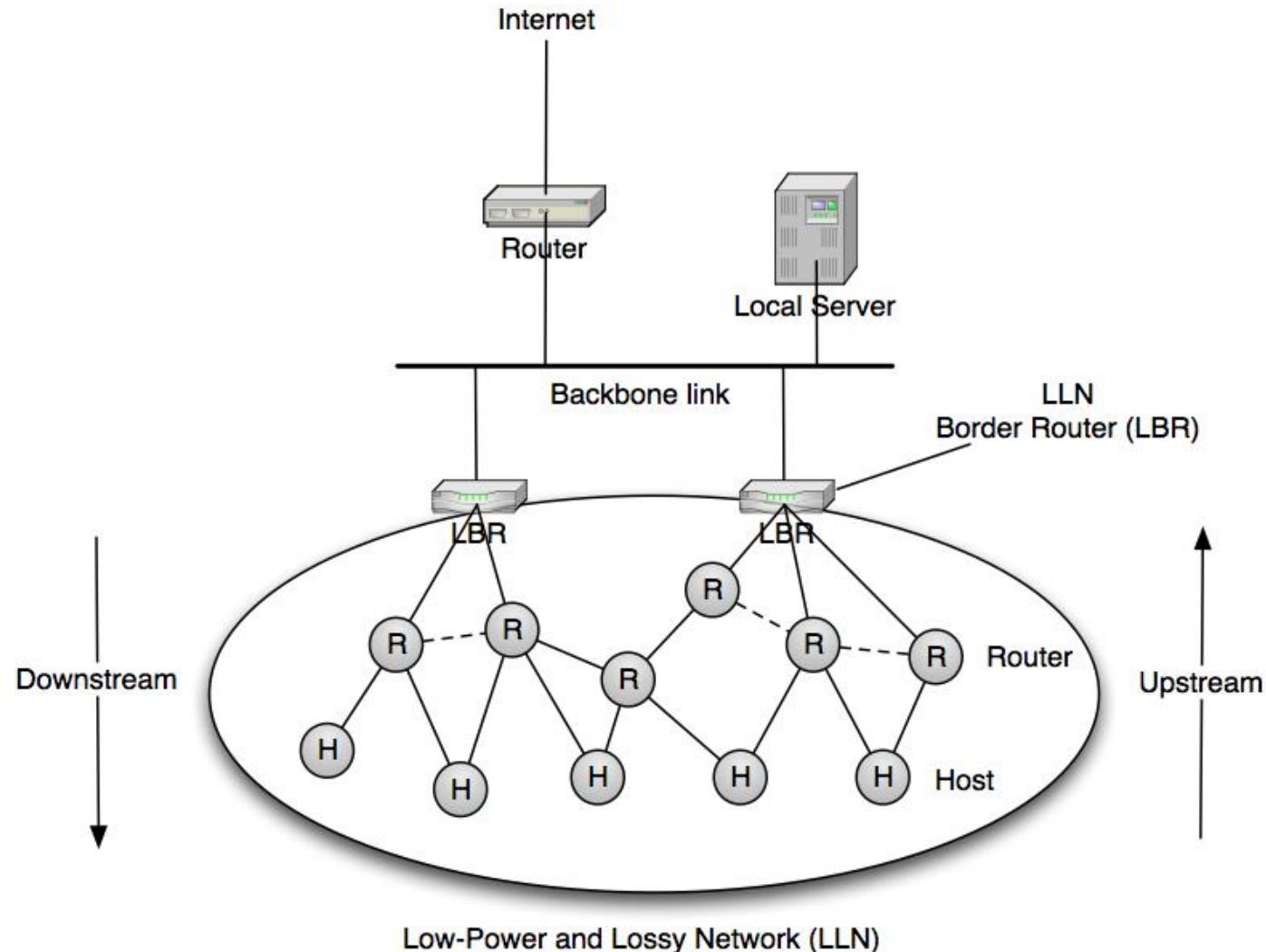
- ▶ 6LoWPAN, como IP, es agnóstico al protocolo de encaminamiento usado
 - ▶ *IP Routing* en nivel 3
 - ▶ *Forwarding* en función de las tablas de enrutamiento
- ▶ Consideraciones especiales para routing en LoWPANs
 - ▶ Única interfaz de *routing*, topología plana
 - ▶ Redes inalámbricas *Low-power and lossy*
 - ▶ Enrutamiento multi-salto



6LoWPAN - Encaminamiento

- ▶ *Routing Over Low power and Lossy networks (ROLL)*
 - ▶ Grupo de trabajo del IETF
- ▶ Protocolo RPL “Ripple”
 - ▶ *IPv6 Routing Protocol for Low-Power and Lossy Networks*
 - ▶ RFC 6550
 - ▶ Algoritmo vector distancia y proactivo
 - ▶ Cada *router* envía información a sus vecinos sobre toda la red, para calcular el camino más corto al destino
 - ▶ Utiliza varias métricas. Diferentes función objetivo
 - ▶ Detección de inconsistencias: evitar bucles, mantener convergencia, etc.
- ▶ Otros protocolos ad-hoc/oportunos: AODV, OLSR, BATMAN, JOKER...

ROLL RPL “Ripple”



6LoWPAN - Seguridad

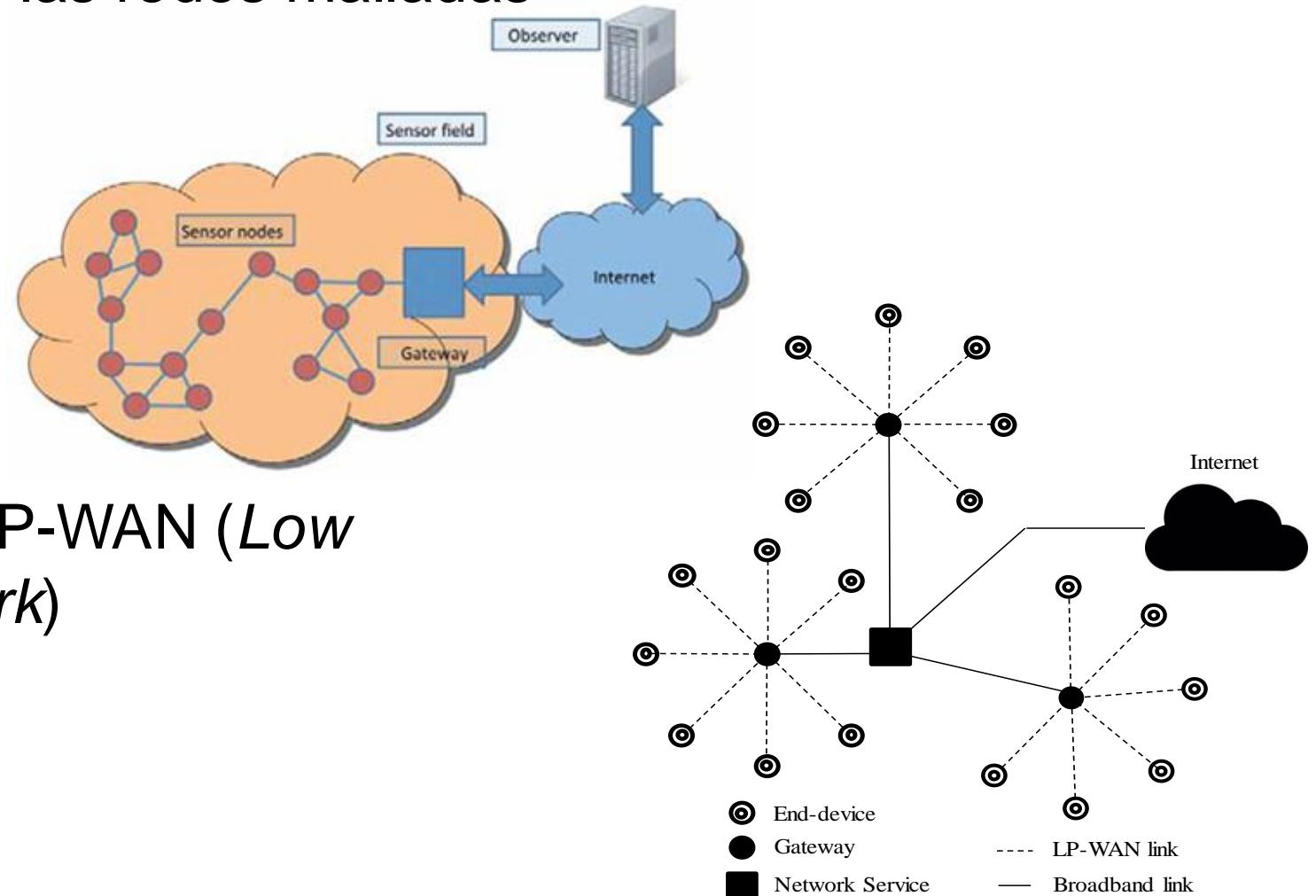
- ▶ En redes inalámbricas el canal es muy vulnerable
 - ▶ Es fácil de esnifar comunicaciones y realizar ataques
- ▶ **Nivel 2:** mecanismos de IEEE 802.15.4
 - ▶ Basado en 128-bit Advanced Encryption Standard (AES)
 - ▶ Proporciona confidencialidad (cifrando) e integridad
 - ▶ Muchos dispositivos incluyen chips para procesamiento AES-128
- ▶ **Nivel 3: IPSec standard [RFC4301]** seguridad IP, *end-to-end*
 - ▶ Dos formatos de cabecera
 - ▶ Authentication Header (AH) in [RFC4302]
 - Sólo integridad y autenticación
 - ▶ Encapsulating Security Payload (ESP) [RFC4303] (mas usado)
 - También cifra para conseguir confidencialidad
 - ▶ Un modo de ESP se define usando AES-CCM (obligatorio en 802.15.4) [RFC4309]
 - ▶ Sirve para el uso en nodos 6LoWPAN
 - ▶ El mismo chip a nivel 2 de IEEE 802.15.4 se puede reusar

Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

LP-WAN

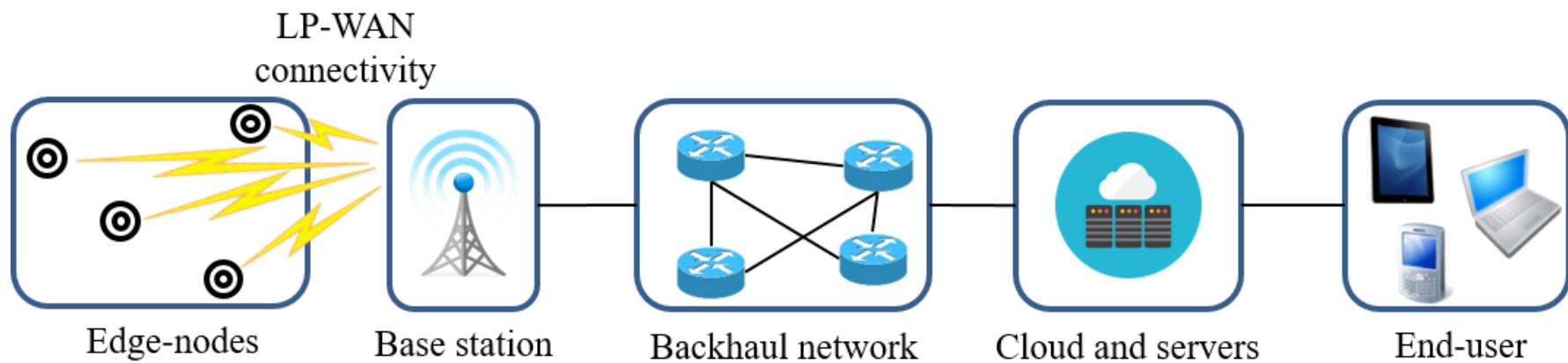
- ▶ Problemas de gestión de las redes malladas de corto alcance
 - ▶ Enrutamiento
 - ▶ Conexión a Internet
 - ▶ Escasa cobertura
- ▶ Cambio de paradigma: LP-WAN (*Low Power-Wide Area Network*)



LP-WAN

▶ Características

- ▶ Largo alcance como las redes celulares (o más)
- ▶ Bajo consumo como las redes de sensores
- ▶ Alta escalabilidad
- ▶ Topología en estrella o estrella de estrellas
- ▶ Uso de frecuencias libres ISM sub-GHz (868 MHz o 902 MHz)
- ▶ Dispositivos de bajo coste
- ▶ Enlaces asimétricos: mayor valor al *uplink*
- ▶ *Roaming*: conexión del dispositivo a distintas estaciones base, ej. control de mercancías



Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

Sigfox



▶ Protocolo propietario

- ▶ Modulación *Ultra Narrow Band* (200 Hz). *Differential Binary Phase Shift Keying* (DBPSK)
- ▶ Tasa de transmisión muy limitada: 100 bps
- ▶ Utilización de bandas libres ISM (*Industrial, Scientific and Medical*) sub-GHz: 868 MHz (Europa) y 902 MHz (EEUU)



Largos alcances y penetración (10 km en campo abierto y 2-3 km en zona urbana) y muy bajo consumo de energía

Sigfox

▶ Limitaciones técnicas muy importantes:

- ▶ Tiempos de transmisión muy altos (2-3 s/msg) + acceso al medio tipo ALOHA + saturación de bandas libres → Límite de mensajes: 140 mensajes al día (*duty cycle*)
- ▶ Tamaño máximo de *payload*: 12 bytes
- ▶ *Downlink* muy limitado (se habilitó en una actualización reciente). Se abre ventana de recepción tras transmitir un mensaje: ahorro energético
- ▶ Seguridad: no ofrece ni cifrado ni seguridad extremo a extremo. Se asume que sólo el usuario conoce el contenido y significado del *payload*. Saltos en frecuencia
- ▶ Éxito: modelo de negocio
 - ▶ Muy **buena cobertura** y escalabilidad (alto número de dispositivos por estación base)
 - ▶ **Amplio despliegue**: varios países europeos cubiertos, mediante convenios con operadores de servicios móviles. España: Cellnex (antigua Abertis Telecom)
 - ▶ Instalación de sistema muy simple. Usuario sólo encargado de dispositivos finales
 - ▶ **Alta eficiencia energética**: las baterías duran más de 10 años
 - ▶ Gestión y presentación automática de los datos: red *backhaul* y gestión en la nube proporcionadas por Sigfox

Sigfox



Estaciones base Sigfox en España



Sigfox - Backend

SIGFOX

- SITE
- BASE STATION
- DEVICE**
- DEVICE TYPE
- USER
- GROUP
- RADIO PLANNING
- BILLING

Information

Location

Messages

Events

Statistics

Event Configuration

Device 1BC1D - Messages

[Purge all messages](#)

From date

To date

Type



page 1

Time	Delay (s)	Header	Data / Decoding	Location	Base station	RSSI (dBm)	SNR (dB)	Freq (MHz)	Rep	Callbacks
2015-08-20 20:27:32	1.1	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-122.90	 12.94	868.1888	2	
					2108	-136.40	 8.18	868.1890	1	
2015-08-20 20:23:32	1.4	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-124.90	 17.26	868.1919	1	
2015-08-20 20:16:26	1.8	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-124.90	 17.89	868.1813	3	
2015-08-20 19:59:07	1.6	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-122.90	 18.26	868.1875	2	
2015-08-20 19:56:57	1.4	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-124.90	 12.19	868.1919	2	
2015-08-20 19:56:08	1.3	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-122.90	 17.83	868.1875	2	
2015-08-20 19:55:56	1.3	0000	41424344454647484950 ASCII: ABCDEFGHIP		210A	-123.90	 17.54	868.1824	2	
2015-08-20 19:27:08	1.2	0000	41424344454647484950		210A	-120.90	 7.71	868.1938	2	

Copyright © SIGFOX - 4.7.3 - 206 - [Terms and conditions](#)

PROTÓCOLOS DE COMUNICACIONES PARA IoT

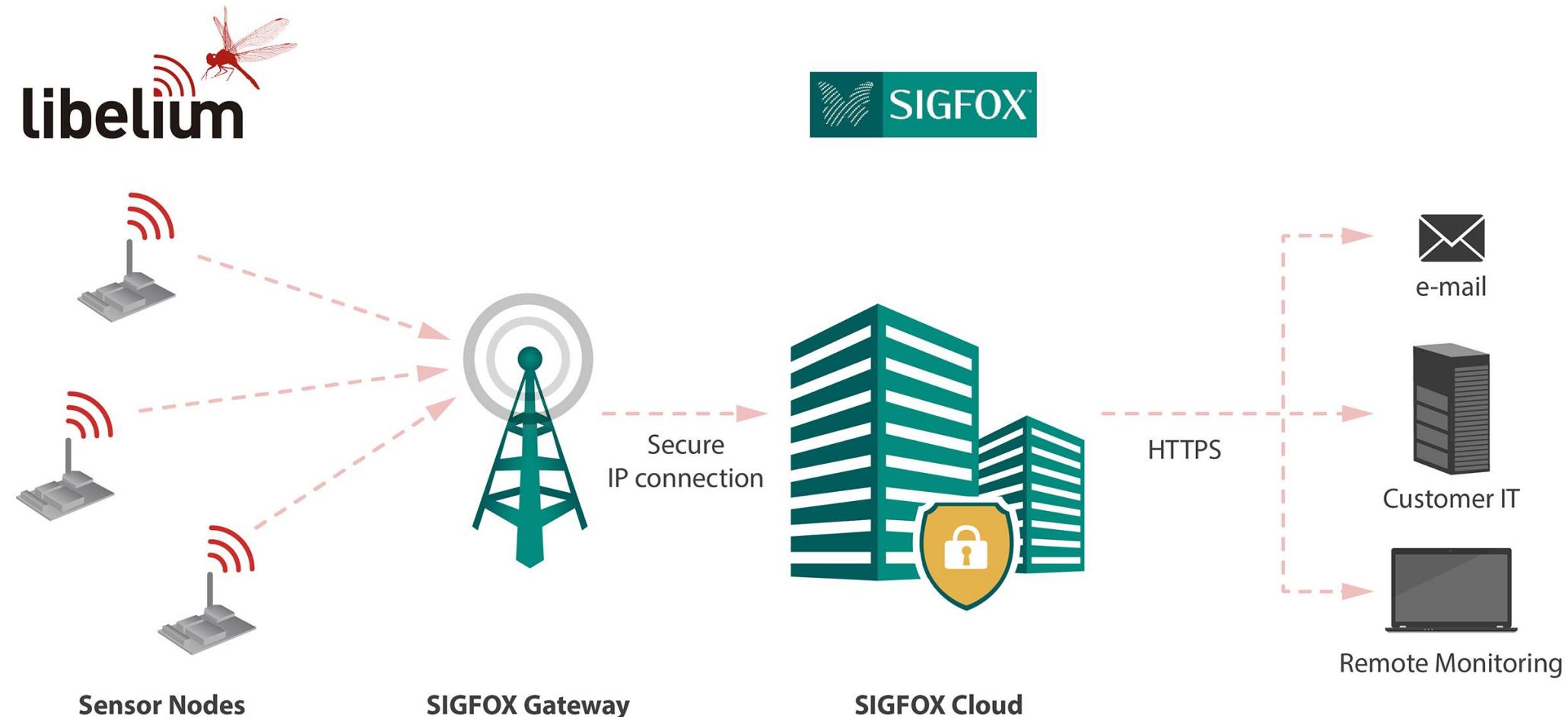
U

MURCIA

17

Sigfox – Caso de uso

- Gestión de alarmas en tiempo real



Sigfox – Caso de uso

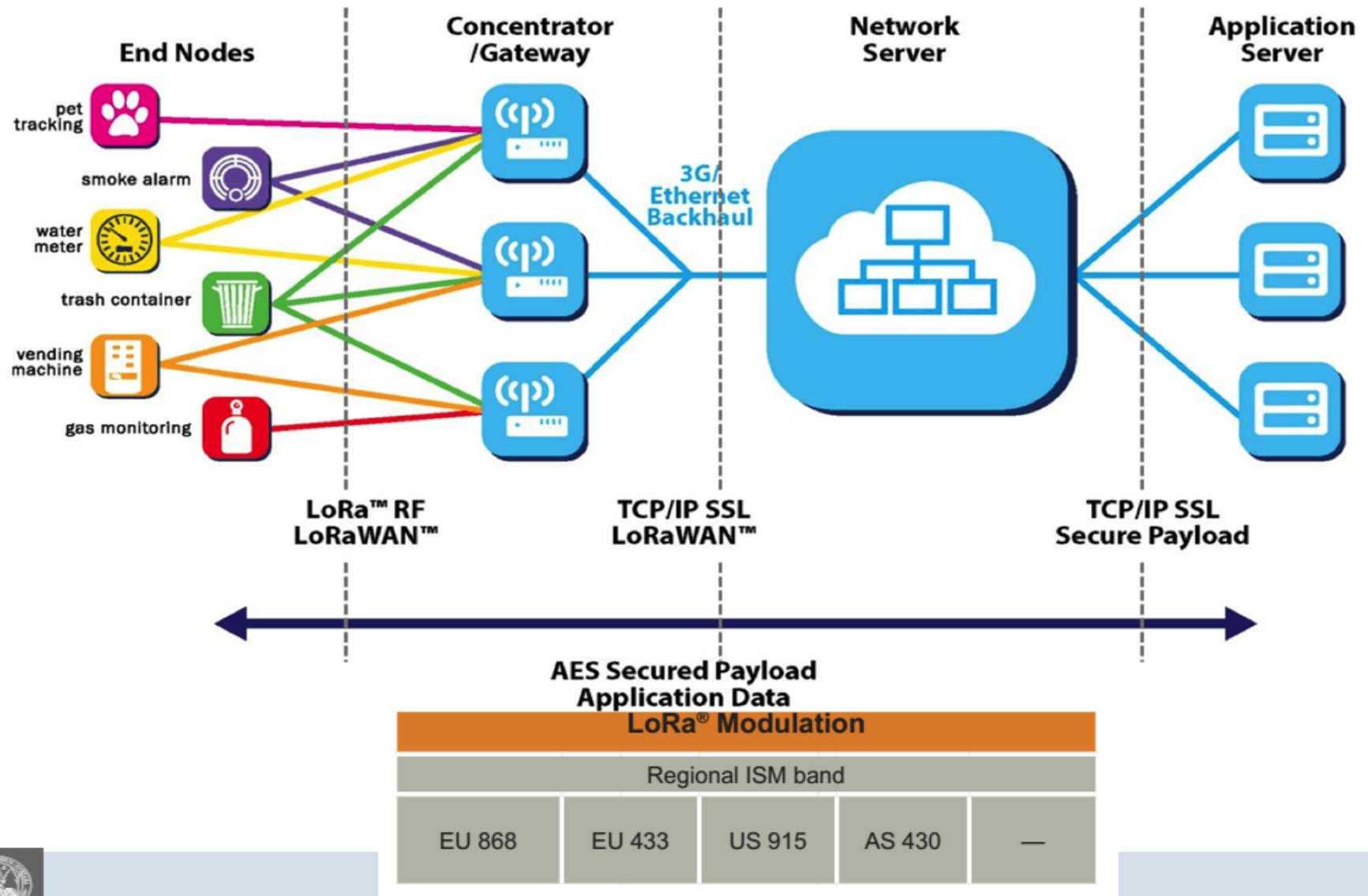
- ▶ Alarmas y sensado en tiempo real



Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

LoRaWAN



LoRaWAN

- ▶ Proporciona conectividad de largo alcance (similar a Sigfox), tanto en *uplink* como en *downlink*:
 - ▶ Nodos Clase A: después de cada transmisión en *uplink*, el nodo abre dos ventanas de escucha para recibir: menor consumo energético
 - ▶ Nodos Clase B: misma funcionalidad que Clase A, pero además, los nodos abren unas ventanas de escucha de forma programada: consumo energético medio
 - ▶ Nodos Clase C: ventana de escucha siempre abierta: dispositivos conectados a alimentación
- ▶ LoRa: Long Range. Modulación (nivel físico) propietaria
- ▶ LoRaWAN: pila de protocolos superior (abierta)

LoRaWAN

- ▶ LoRa: modulación de espectro ensanchado
- ▶ Parámetros de configuración
 - ▶ *Spreading Factor (SF)*: 7-12
 - ▶ *Coding Rate (CR)*: “4/5”, “4/6”, “4/7” y “4/8”
 - ▶ Tamaño de paquete: 51 – 222 Bytes
 - ▶ Ancho de banda: 125 or 250 kHz
- ▶ Estos factores influyen en el tiempo en el aire de la trama: diferentes *duty cycles* (mensajes diarios), según configuración empleada
- ▶ A mayor SF y CR, mayor tiempo en el aire, pero comunicación más segura: *time-on-air* ↔ *fiabilidad*

LoRaWAN

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8				
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kbps				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

In definition by Technical Committee

LoRaWAN



Sigfox vs. LoRaWAN

- ▶ LoRaWAN es una solución tecnológicamente más avanzada
- ▶ LoRaWAN presenta mejores prestaciones y parámetros de configuración adaptables a los distintos entornos de transmisión
- ▶ Sigfox tiene una arquitectura de red más amigable para el usuario final que no quiere invertir en infraestructura ni sabe como gestionar sus datos en la nube

LoRaWAN – Caso de uso

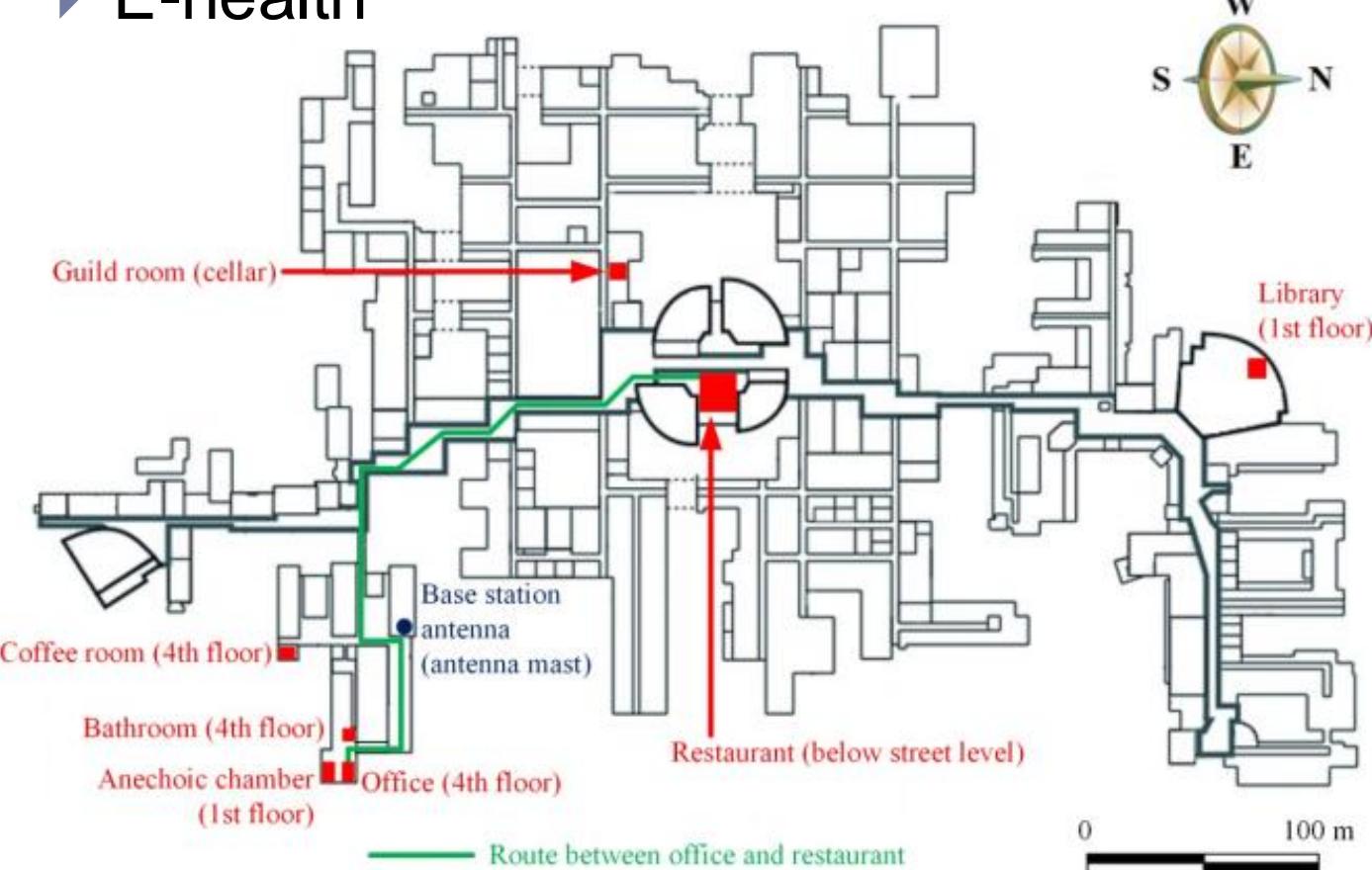
► E-health



J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, Worcester, MA, 2016, pp. 1-5.

LoRaWAN – Caso de uso

▶ E-health

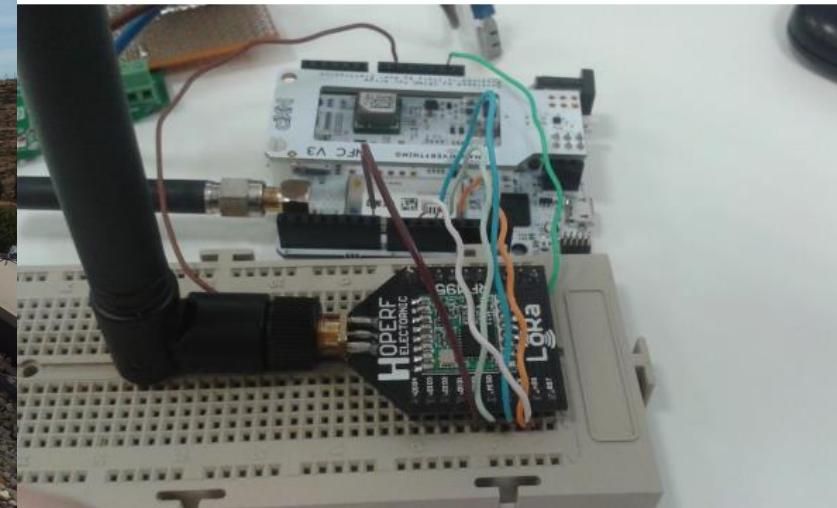


Location	Distance to the BS	No. of Tx packets	No. of Rx packets	Success ratio
Office	65±5 m	1796	1758	97.9 %
Bathroom	54±2 m	331	329	99.4 %
Coffee room	52±5 m	736	717	97.4 %
Restaurant	180±30 m	1245	1193	95.8 %
Library	390±30 m	878	831	94.7 %
Anechoic chamber	68±15 m	291	0	0 %
Guild room	195±15 m	340	322	94.7 %
Total (without anechoic chamber)	-	5326	5150	96.7 %

J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, Worcester, MA, 2016, pp. 1-5.

LoRaWAN – Caso de uso

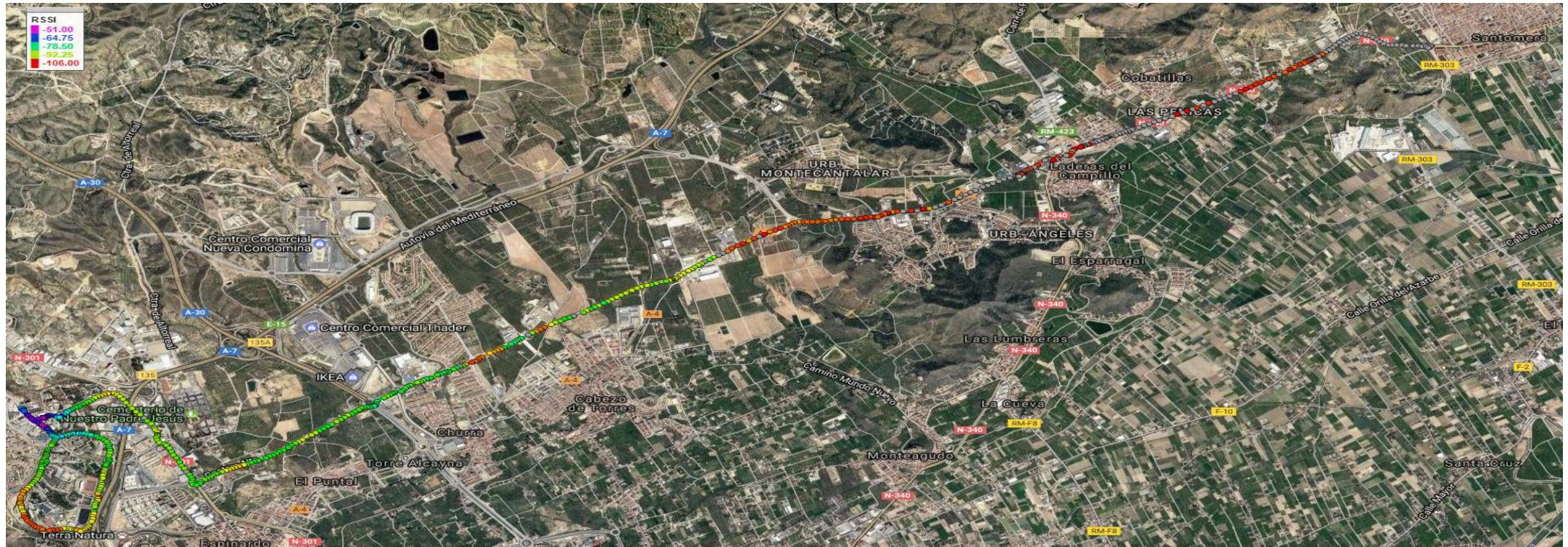
▶ Transmisiones a larga distancia



J. Sanchez-Gomez, R. Sanchez-Iborra, A. Skarmeta, "Experimental comparison of LoRa and FSK as IoT-communication-enabling modulations," *IEEE GlobeCom '17*, Singapur, 2017.

LoRaWAN – Caso de uso

▶ Transmisiones a larga distancia



J. Sanchez-Gomez, R. Sanchez-Iborra, A. Skarmeta, "Experimental comparison of LoRa and FSK as IoT-communication-enabling modulations," *IEEE GlobeCom '17*, Singapur, 2017.

Índice

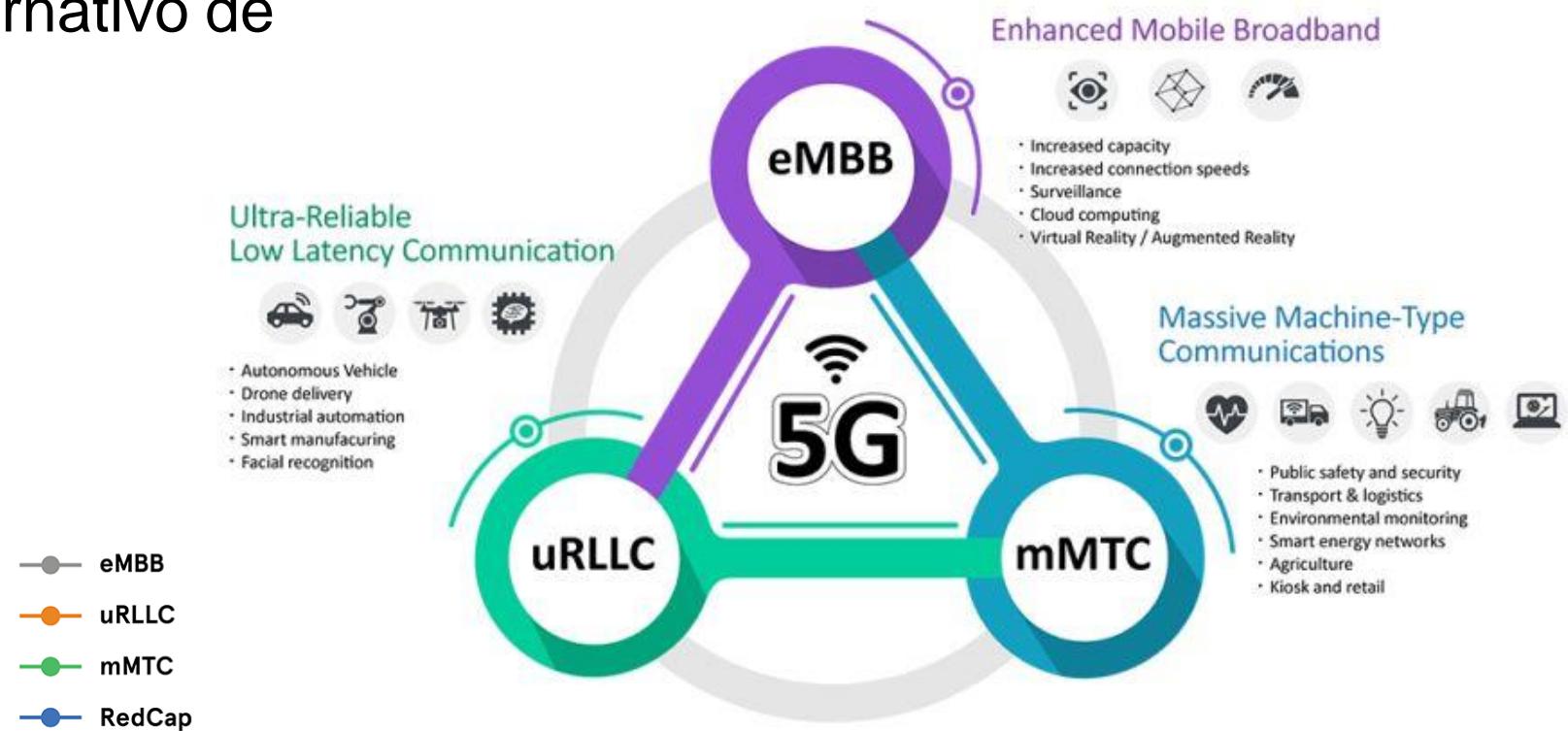
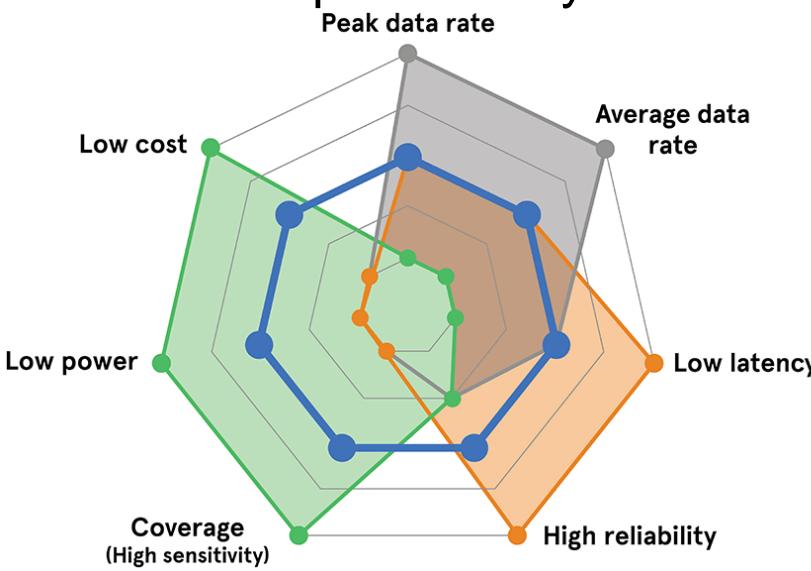
- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

HaLow, BLE, NB-IoT y CAT-M1

- ▶ "Nuevos" contendientes en el mundo IoT.
- ▶ BLE deriva de Bluetooth "Bluetooth Low Energy"
 - 1 y 2 Mbps, hasta 100m
 - LE-Coded, 125 y 500kbps, varios km – casi se puede considerar LP-WAN
- ▶ HaLow – 802.11ah
 - Familia de los protocolos WiFi, en banda libre sub-Ghz
 - Baja la tasa a ~8mbps de máximo ganando distancia de transmisión (~1km).
- ▶ CAT-M1 y NB-IoT (predecesores de 5G MMTC)
 - Estándares celulares de la familia de 3GPP (4G)
 - Basados en 4G, diseñados para un módem más sencillo y de menor consumo
 - Usan la red de telefonía y alcanzan ~60kbps (NB-IoT) y ~1Mbps (CAT M1)

HaLow, BLE, NB-IoT y CAT-M1

- ▶ Z-Wave: propietario, alternativa a ZigBee
- ▶ ANT, ANT+: propietario, alternativo de Garmin a BLE
- ▶ 5G MMTC / RedCap
 - "not quite there yet"



Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

CoAP

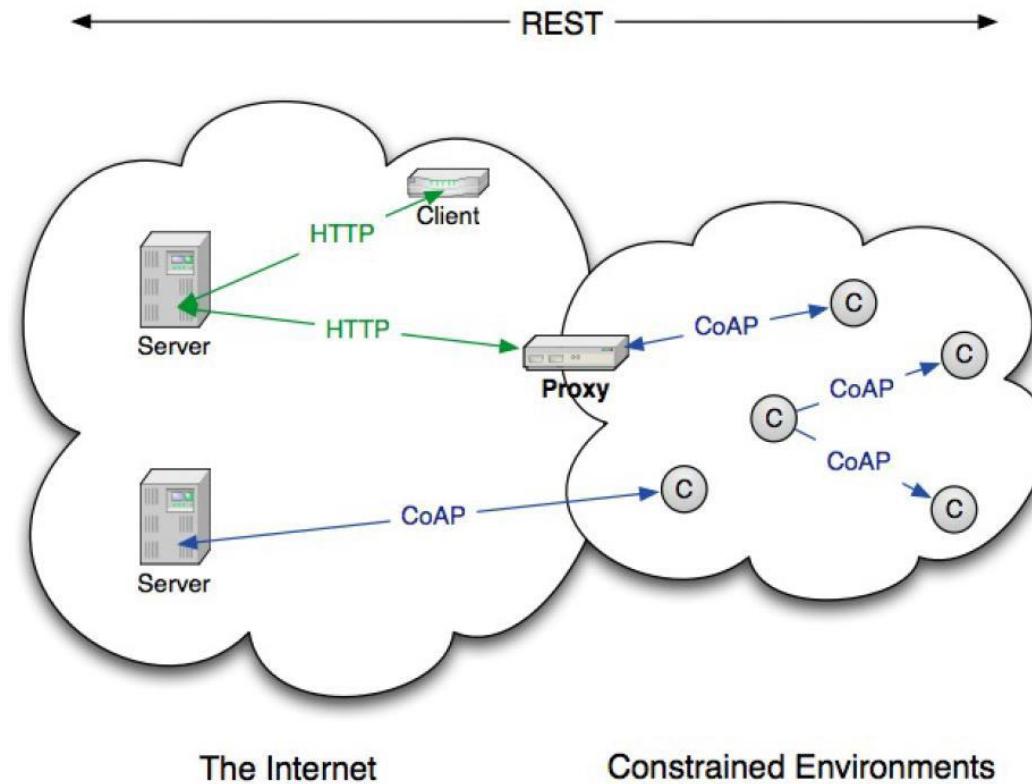


▶ CoAP – Constrained Application Protocol

- ▶ Desarrollado en el CoRE Working Group (Constrained Resource Environments) del IETF
- ▶ Definido
 - ▶ RFC 7252 (2014)
 - ▶ RFC 7641: extensión OBSERVE (2015)
- ▶ Autores:
 - ▶ Z. Shelby (Sensinode, ARM)
 - ▶ K. Hartke, C. Bormann (Uni Bremen TZI)
- ▶ CoAP es un protocolo eficiente modelo cliente-servidor basado en RESTful
 - ▶ Soporta operaciones GET / PUT / POST / DELETE (como HTTP)
 - ▶ Aparentemente sencillo, pero muy potente

CoAP

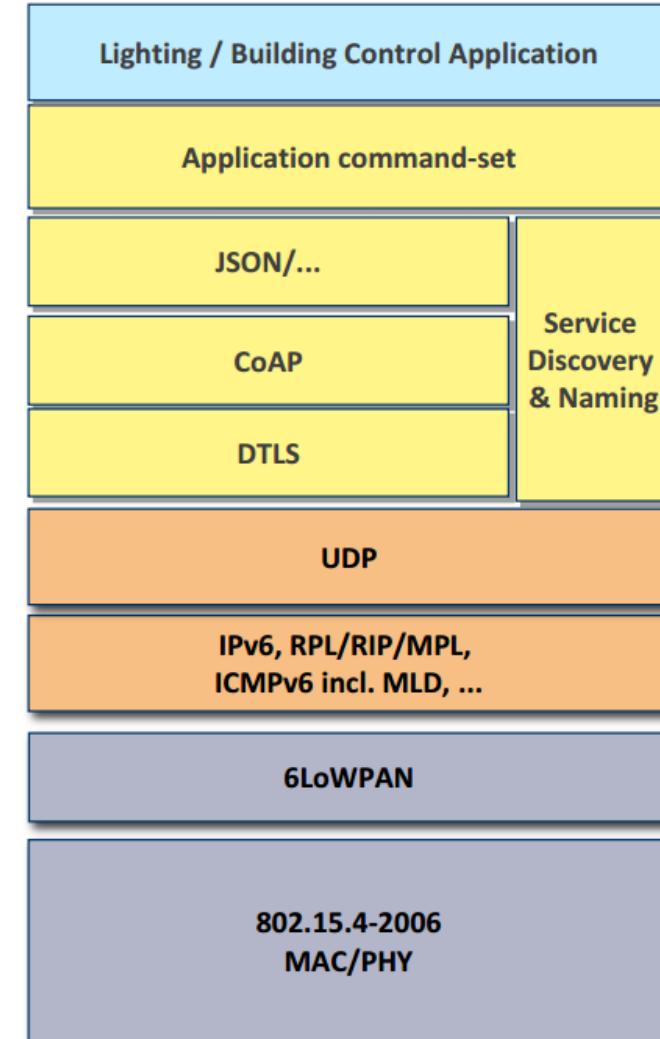
The CoAP Architecture



	Traditional IP	IoT protocols
Application protocol	HTTP (and related protocol, eg SMTP)	CoAP
Transport layer	TCP (or UDP)	UDP only
Network layer	IPv4 / IPv6	6LoWPAN
Link layer	802.11n (or ethernet)	802.15.4e

CoAP

- ▶ Usuarios CoAP:
 - ▶ Usuarios de servicios web
 - ▶ CoAP implementa un protocolo para servicios web
 - ▶ Otros dispositivos CoAP (máquinas)
 - ▶ Intercambio de información entre dispositivos finales
 - ▶ Servidores de gestión
 - ▶ Ej. uso de LWM2M, para acceder y gestionar un dispositivo restringido usando CoAP
- ▶ CoAP es un rediseño (no una simple compresión) de HTTP
 - ▶ CoAP puede ser traducido a HTTP para interoperabilidad e integración con la WEB
- ▶ CoAP se ha definido sobre UDP como protocolo de transporte
 - ▶ Puerto 5683
 - ▶ 5684 para CoAP sobre DTLS (Datagram Transport Layer Security)



CoAP

- ▶ Emplea recursos limitados:
 - ▶ Tamaño mensajes reducido (cabecera 4 bytes)
 - ▶ Redes *constrained*, ej. Low-Power and Lossy Networks (LLN)
 - ▶ Soporta nodos inactivos, uso de *proxies*
- ▶ Reduce las ineficiencias de operaciones REST
 - ▶ No codifica en texto plano, y reduce el tamaño de los mensajes
 - ▶ No utiliza TCP que añade overhead, sino UDP (no orientado a conexión)
- ▶ Permite operaciones *Machine to Machine* (M2M)
 - ▶ Descubrimiento de recursos
 - ▶ publicación / suscripción / notificación
 - ▶ Negociación de contenidos: distintas representaciones
 - ▶ *multicast*

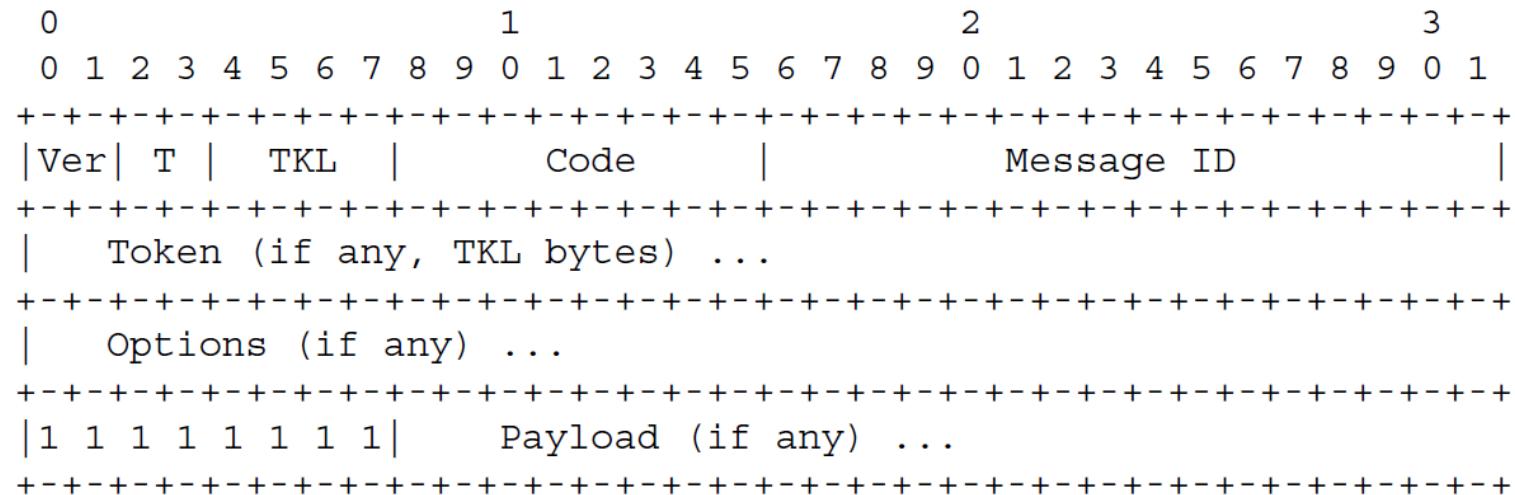
CoAP

▶ Mensajes CoAP

- ▶ Se utiliza UDP (no orientado a la conexión, no garantía entrega)
- ▶ Control de entrega realizado a nivel de aplicación por el protocolo CoAP. Cada mensaje se marca como “confirmable” o “no confirmable”:
 - ▶ Mensajes confirmables: requiere de un ACK
 - ▶ Mensajes no-confirmables: *fire and forget*
- ▶ Seguridad → DTLS (Datagram Transport Layer Security): similar a TLS con soporte para RSA, AES y ECC

CoAP

▶ Formato del mensaje



- ▶ **type** → indica el rol del mensaje como parte de la transacción. CON / ACK / NON / RST
- ▶ **TKL** → token length
- ▶ **code** → da información adicional sobre el propósito del mensaje:
 - ▶ Request o response. GET, POST, PUT, DELETE
- ▶ **Message id** → valor único a la transacción
- ▶ **token** → para especificar el concepto de “topic”
- ▶ **options** → para incluir parámetros y mecanismos de gestión de mensajes

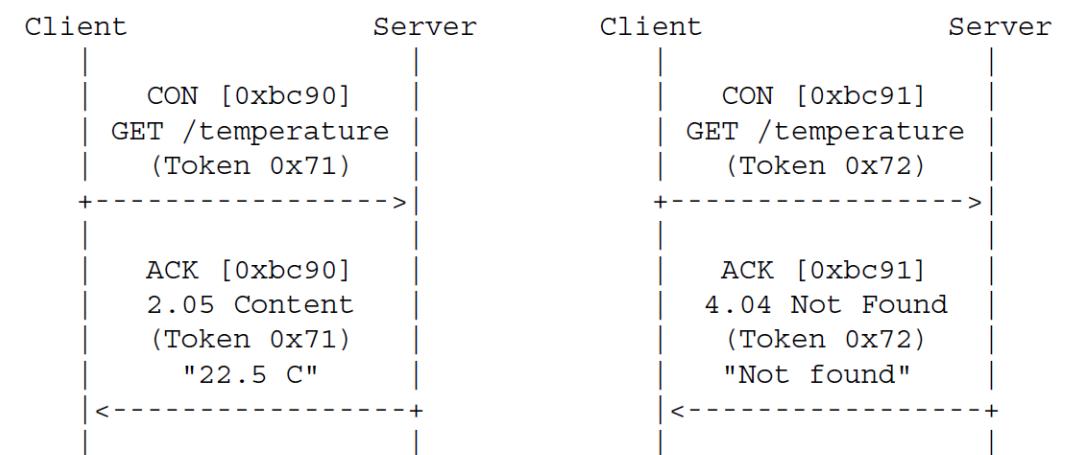
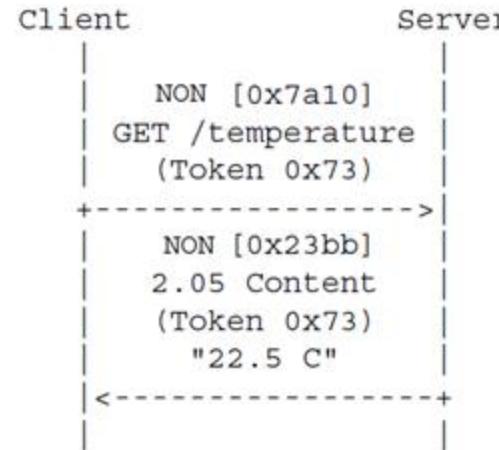
CoAP

▶ Tipos Mensajes

- ▶ CON → Confirmable
- ▶ NON → Non-Confirmable
- ▶ ACK → acknowledge CON +piggyback
- ▶ RST→ reset interaction

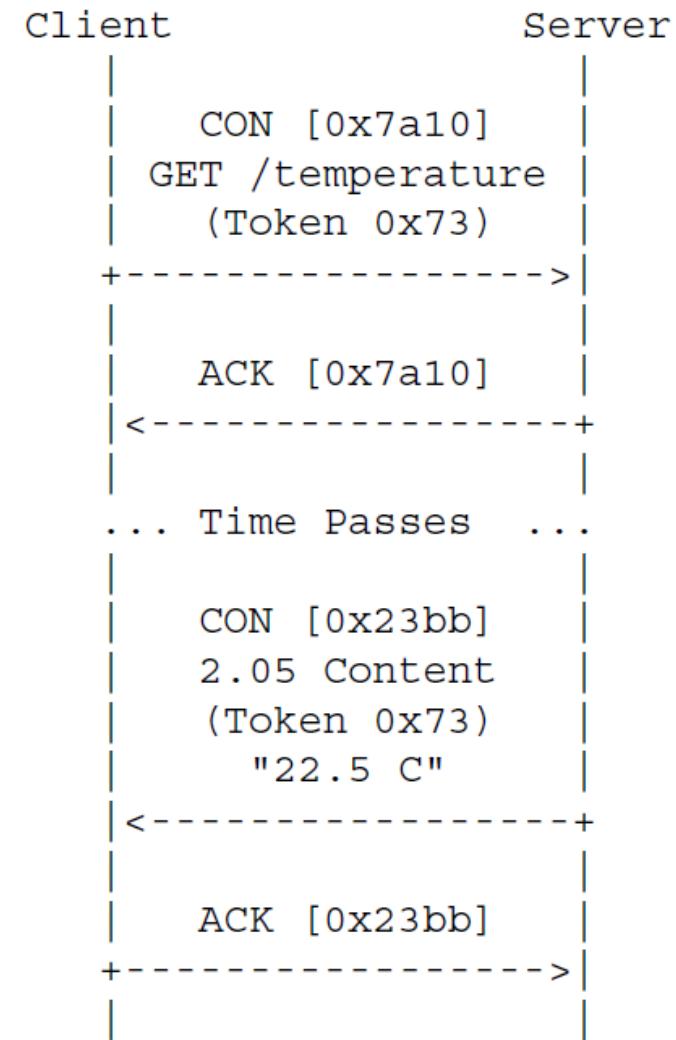
▶ Ej. Mensaje no-confirmable

▶ Ej. Mensajes confirmables con ACK *piggybacked*



CoAP

- ▶ Respuesta confirmable asíncrona:
 - ▶ El ACK y la respuesta se envían en mensajes distintos



CoAP

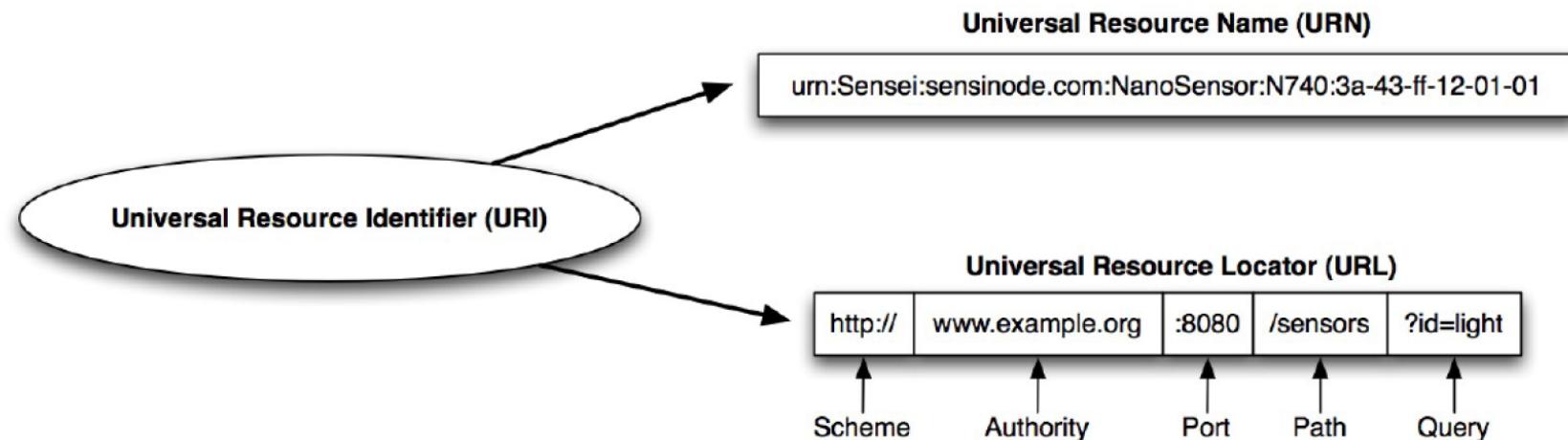
- ▶ **Code format “c.dd”**
- ▶ Class 0 (request)
- ▶ Class 2 (success)
- ▶ Class 4 (client error)
- ▶ Class 5 (server error)

Code	Name	Reference
0.01	GET	[RFC7252]
0.02	POST	[RFC7252]
0.03	PUT	[RFC7252]
0.04	DELETE	[RFC7252]

Code	Description	Reference
2.01	Created	[RFC7252]
2.02	Deleted	[RFC7252]
2.03	Valid	[RFC7252]
2.04	Changed	[RFC7252]
2.05	Content	[RFC7252]
4.00	Bad Request	[RFC7252]
4.01	Unauthorized	[RFC7252]
4.02	Bad Option	[RFC7252]
4.03	Forbidden	[RFC7252]
4.04	Not Found	[RFC7252]
4.05	Method Not Allowed	[RFC7252]
4.06	Not Acceptable	[RFC7252]
4.12	Precondition Failed	[RFC7252]
4.13	Request Entity Too Large	[RFC7252]
4.15	Unsupported Content-Format	[RFC7252]
5.00	Internal Server Error	[RFC7252]
5.01	Not Implemented	[RFC7252]
5.02	Bad Gateway	[RFC7252]
5.03	Service Unavailable	[RFC7252]
5.04	Gateway Timeout	[RFC7252]
5.05	Proxying Not Supported	[RFC7252]

códigos de respuesta

CoAP



▶ CoAP URI

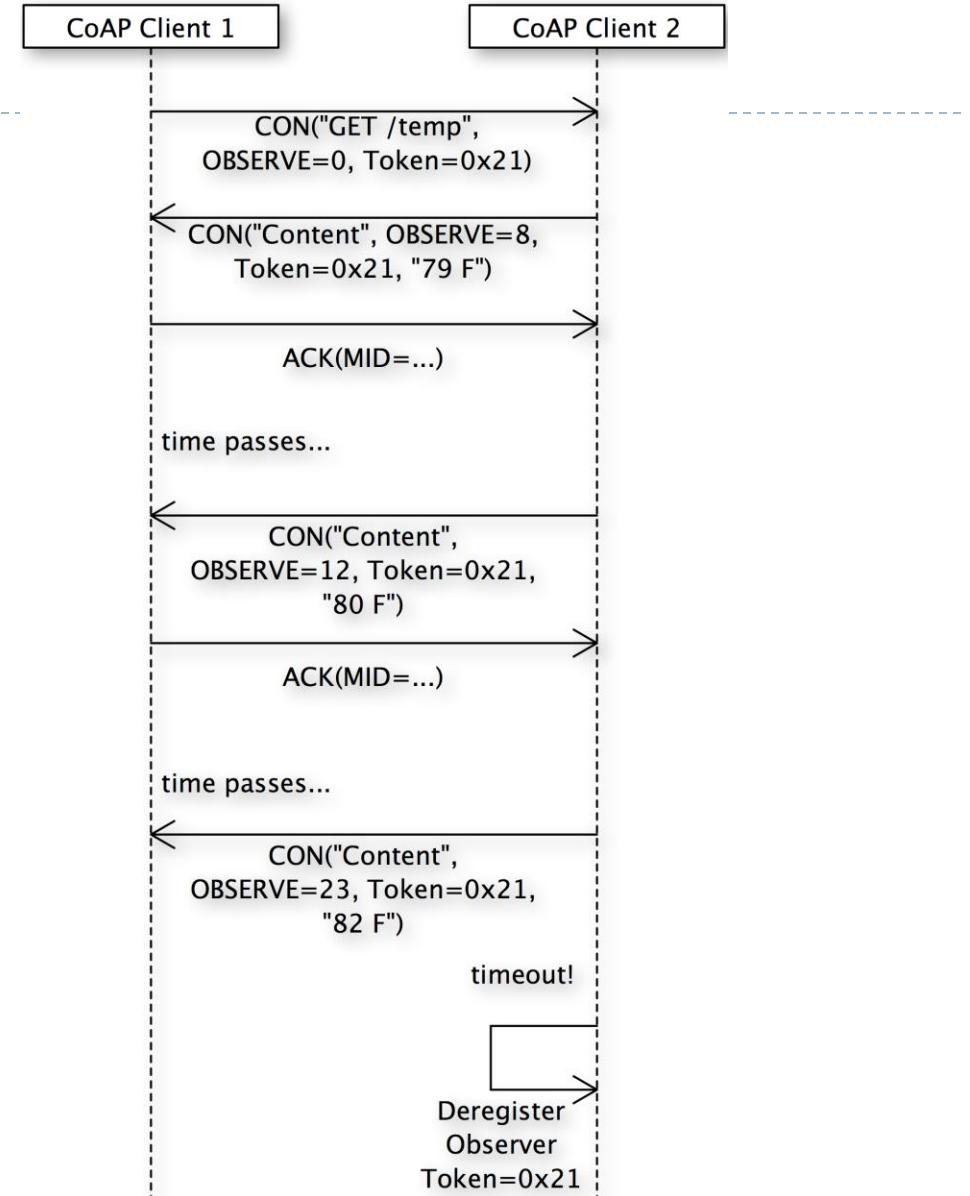
```
coap-URI = "coap:" "://" host [ ":" port ] path-abempty [ "?" query ]
```

▶ Ejemplos

- ▶ `coap://example.inf.um.es:5683/sensors/light1`
- ▶ `coaps://example.inf.um.es:5684/sensors/temp?min=10`

CoAP

- ▶ Opción **Observer**
- ▶ Monitorizar un valor sin realizar *polling* → consumo de recursos
- ▶ El valor de la opción *Observe* en la request
 - ▶ 0 registrar
 - ▶ 1 eliminar de la lista
- ▶ En las notificaciones el valor *Observe* es un contador incremental para ordenar notificaciones
- ▶ Notificaciones son:
 - ▶ Tipo=CON
 - ▶ Code=response

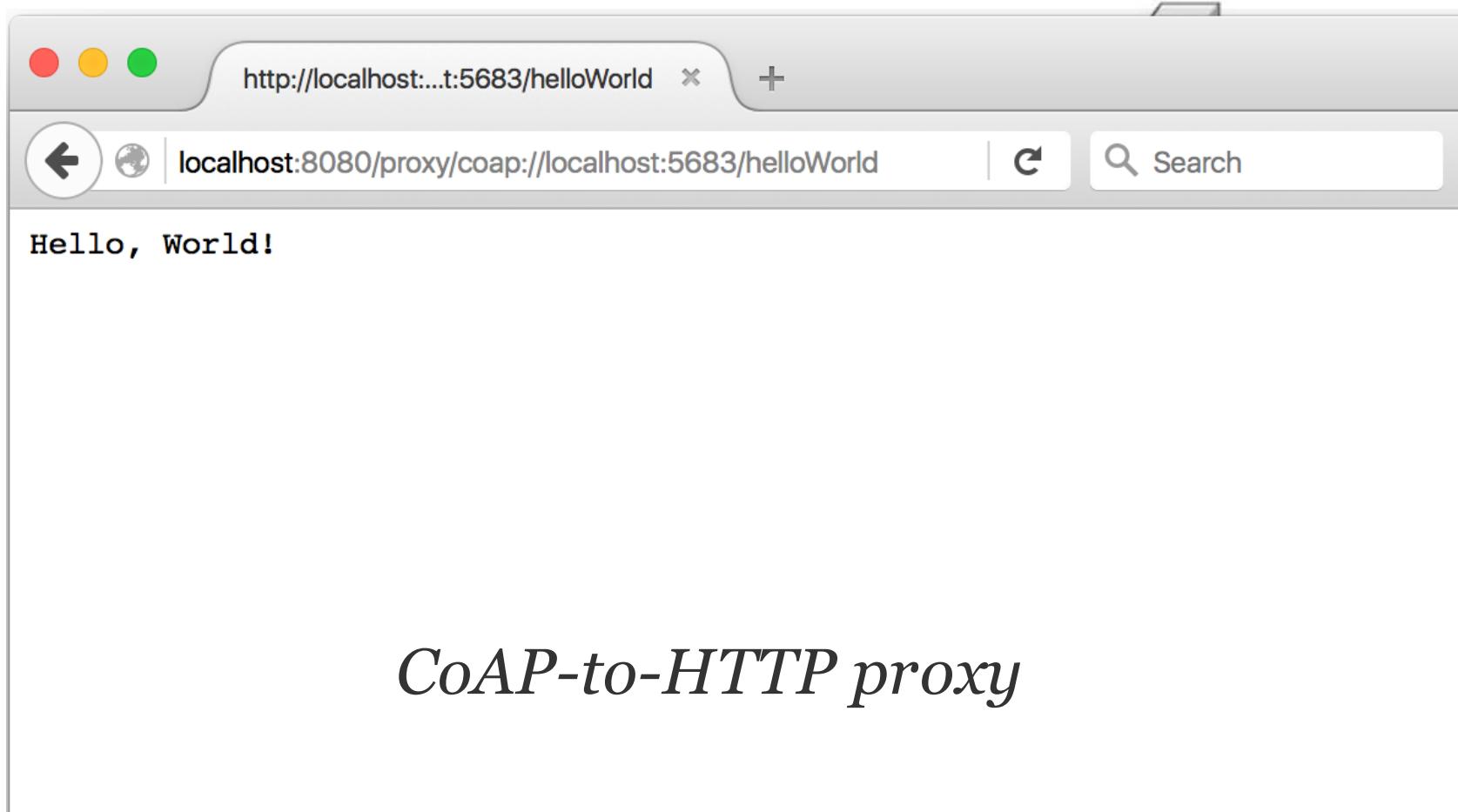


CoAP

- ▶ *Forward proxy*
 - ▶ Actúa como cliente, en nombre de otro cliente
 - ▶ El cliente selecciona el proxy
 - ▶ Puede hacer traducción
- ▶ *Reverse proxy*
 - ▶ Actúa en nombre de otro servidor
 - ▶ Satisface peticiones del servidor para el que actúa, como si fuera el servidor al que quiere acceder el cliente
 - ▶ El cliente puede no saber que está atacando a un reverse proxy
- ▶ Los *proxies* puede cachear mensajes
- ▶ Los *proxies* puede traducir, ej. CoAP a HTTP

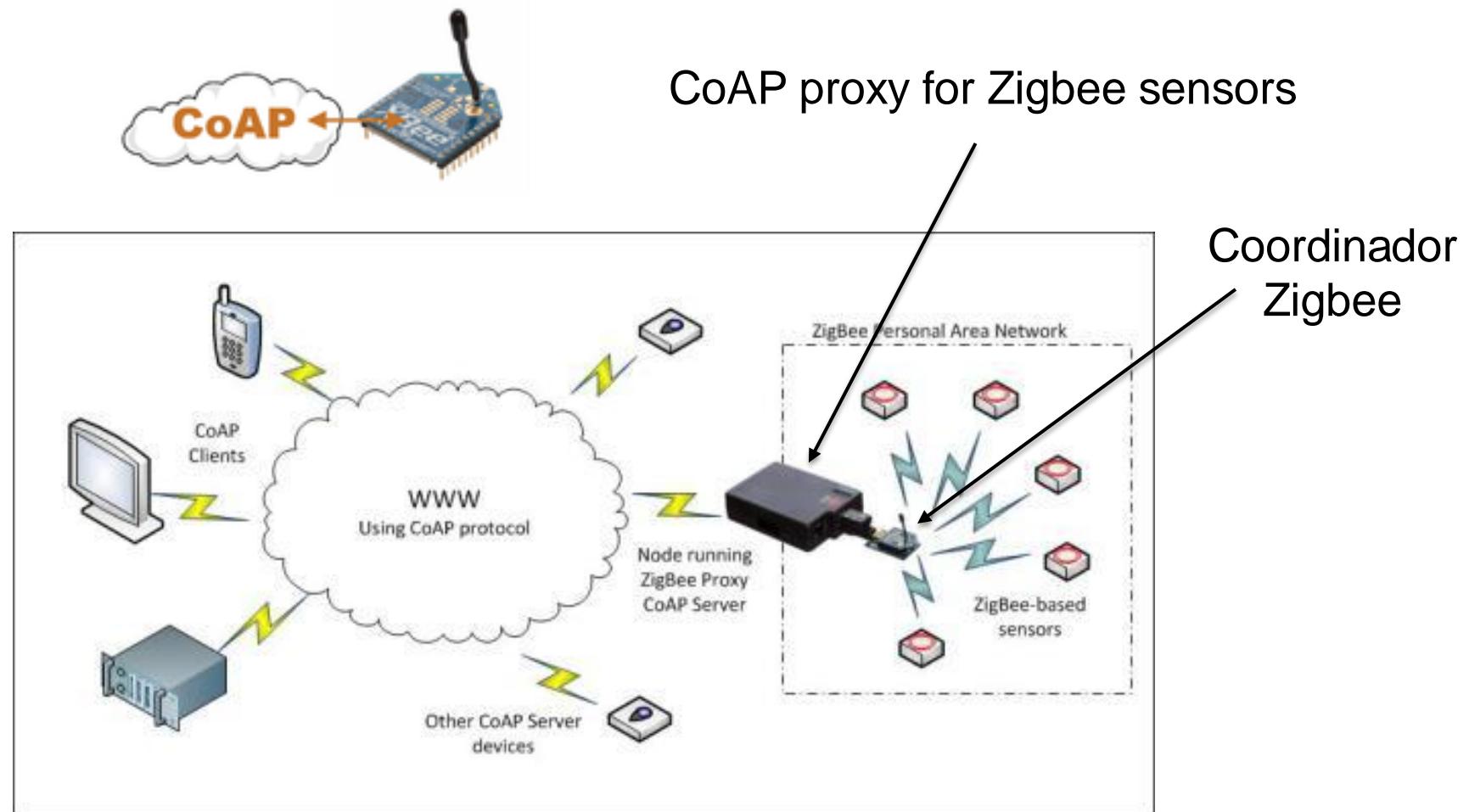
CoAP

▶ Ejemplo Proxy



CoAP - Caso de uso

▶ Integración CoAP-Zigbee



CoAP - Caso de uso

▶ Integración CoAP-Zigbee



```
pi@raspberrypi: ~
File Edit View Search Terminal Help
pi@192.168.2.26's password:
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/**/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul  6 18:17:25 2014 from woodudesk
pi@raspberrypi ~ $ ./ZigBeeCoapProxyServer 192.168.2.26
ID=0013A20040B406EF Temp=11.348974 Light=615.835777 Humidity=38.813166
ID=0013A20040B405F5 Temp=13.812317 Light=638.123167 Humidity=41.156244
```

Índice

- ▶ **Introducción**
- ▶ **IEEE 802.15.4**
 - ▶ Zigbee
 - ▶ 6LoWPAN – IPv6 over low power Wireless Personal Area Networks
- ▶ **LP-WAN**
 - ▶ Sigfox
 - ▶ LoRaWAN – Long Range Wide Area Network
- ▶ **HaLow, BLE, NB-IoT y CAT-M1**
- ▶ **CoAP - Constrained Application Protocol**
- ▶ **MQTT - MQ Telemetry Transport**

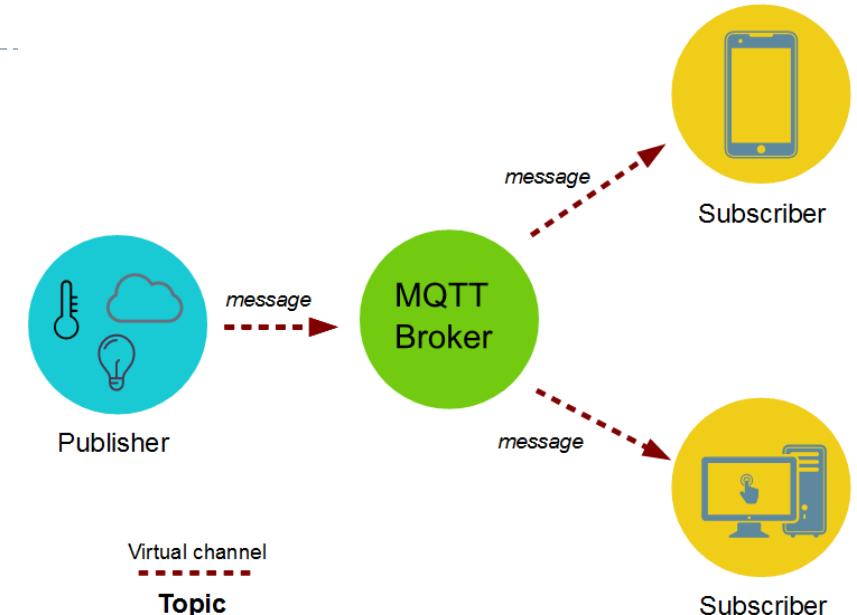
MQTT



- ▶ *Message Queue Telemetry Transport (MQTT)*
 - ▶ Protocolo ligero de publicación/subscripción **sobre TCP/IP** para sensores y dispositivos y redes “constrained”.
- ▶ Estándar de OASIS para IoT (2014), pero diseñado por IBM para conectar instalaciones petrolíferas vía satélite (1999)
- ▶ Ideal para situaciones con comunicaciones M2M e IoT:
 - ▶ Simple de implementar
 - ▶ Provee una capa de QoS
 - ▶ Requiere de poco ancho de banda
 - ▶ Agnóstico a las aplicaciones de capa superior
 - ▶ Permite el establecimiento continuo de una conexión (TCP)
- ▶ Amplio grado de desarrollo y despliegue:
 - ▶ Arduino
 - ▶ Android/iOS
 - ▶ C/C++/C#
 - ▶ Java/JavaScript

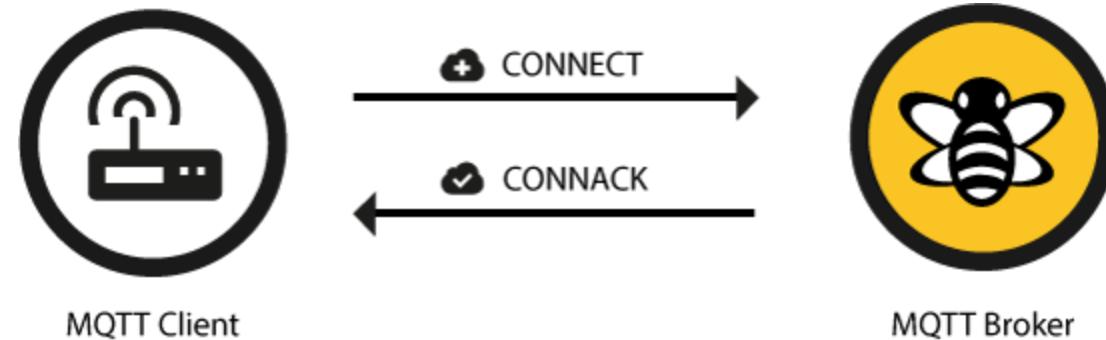
MQTT

- ▶ El productor publica el mensaje solo una vez (al servidor/broker)
- ▶ Múltiples consumidores (aplicaciones/dispositivos) reciben el mensaje a través del broker
- ▶ Desacopla al productor y al consumidor. Espacio/tiempo/sincroniz.
- ▶ El productor manda el mensaje con un *topic*
- ▶ Los consumidores se han suscrito previamente a ese *topic*
- ▶ El servidor/broker realiza la asociación entre publicaciones y suscripciones:
recibe→filtra→asocia→re-envía
 - ▶ Si no hay ninguna asociación, el mensaje se descarta
 - ▶ Si uno o más mensajes cumplen la asociación, el mensaje se entrega al consumidor correspondiente



MQTT

MQTT Connection



MQTT-Packet:

CONNECT



contains:

clientId
cleanSession
username (optional)
password (optional)
lastWillTopic (optional)
lastWillQos (optional)
lastWillMessage (optional)
lastWillRetain (optional)
keepAlive

Example

"client-1"
true
"hans"
"letmein"
"/hans/will"
2
"unexpected exit"
false
60

MQTT-Packet:

CONNACK



contains:

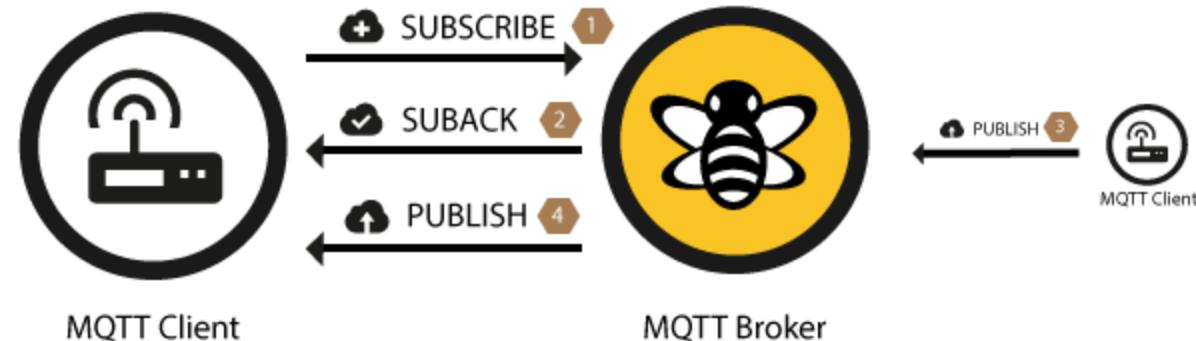
sessionPresent
returnCode

Example

true
0

MQTT

Subscription/publishing



MQTT-Packet:

SUBSCRIBE



contains:
packetId
qos1 } (list of topic + qos)
topic1
qos2 }
topic2
...

Example
4312
1
"topic/1"
0
"topic/2"
...

MQTT-Packet:

SUBACK

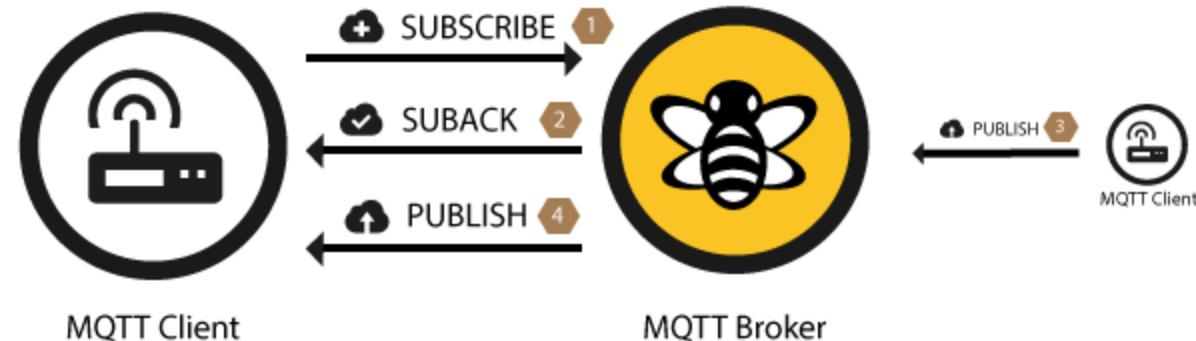


contains:
packetId
returnCode 1 (one returnCode for each topic from SUBSCRIBE, in the same order)
returnCode 2
...
...

Example
4313
2
0
...

MQTT

Subscription/publishing

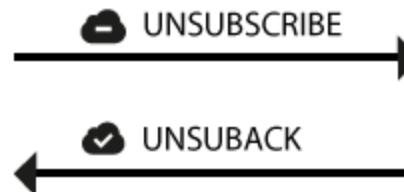


MQTT

Unsubscribe



MQTT Client



MQTT Broker

MQTT-Packet:

UNSUBSCRIBE

contains:

packetId
topic1 } (list of topics)
topic2
...
...

Example

4315

"topic/1"
"topic/2"

...

MQTT-Packet:

UNSUBACK

Example

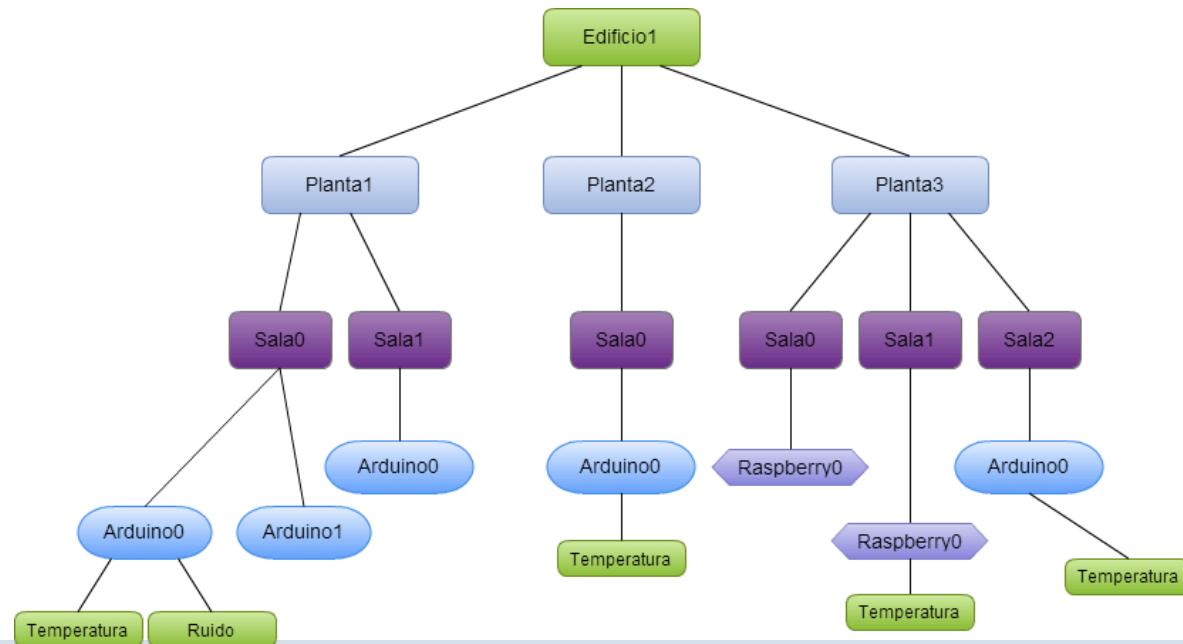
4316

contains:

packetId

MQTT

- ▶ Topics con estructura jerárquica. Cada jerarquía se separa con ‘/’. Por ejemplo:
 - ▶ “edificio1/planta3/sala1/raspberry0/temperatura”
 - ▶ “edificio1/planta1/sala0/arduino0/ruido”
 - ▶ Suscripción agregada (no la publicación) ej.“edificio1/planta2/#”



MQTT

▶ Ejemplo de *topic*:

- ▶ Una casa que publica información sobre si misma:
 - <country>/<region>/<town>/<postcode>/<house>/energyConsumption
 - <country>/<region>/<town>/<postcode>/<house>/fireAlarm
 - <country>/<region>/<town>/<postcode>/<house>/floodingAlarm
- Un suscriptor se puede suscribir a un *topic* concreto valor absoluto o usar *wildcards*
 - ▶ Single-level *wildcards* “+” → puede aparecer en cualquier lugar del nombre del *topic*
 - ▶ Multi-level *wildcards* “#” → deben aparecer al final del *namespace*
 - ▶ Los *wildcards* se deben poner a continuación del separador
 - ▶ No se pueden usar *wildcards* para la publicación
 - ▶ Ejemplos:
 - ▶ Spain/Murcia/Espinardo/30110/1/energyConsumption
 - Consumo de energía para una casa concreta en Espinardo
 - ▶ Spain/Murcia/Espinardo/+/+/energyConsumption
 - Consumo de energía para todas las casas de Espinardo
 - ▶ Spain/Murcia/Espinardo/30110/#
 - Consumo de energía y alarmas (2) para todas las casas con el código postal:30110

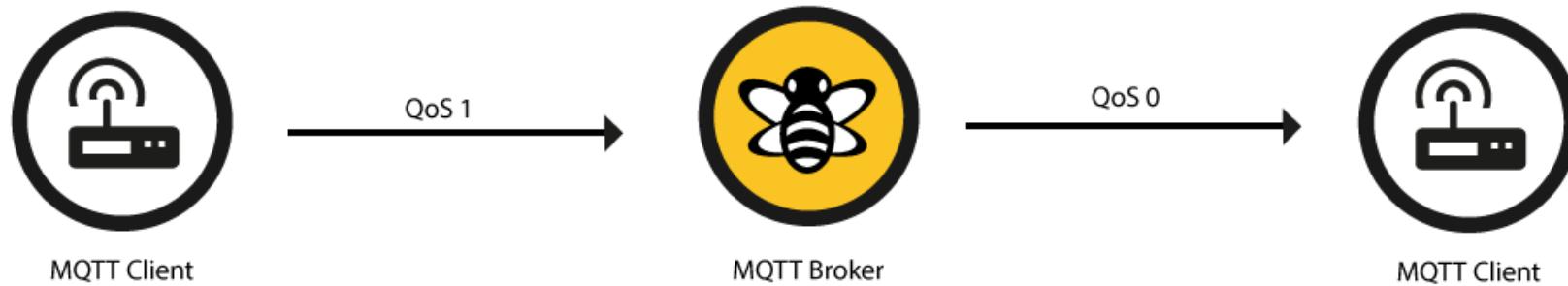


MQTT

- ▶ Diseñado para dispositivos *constrained*:
 - ▶ Recursos limitados en cuanto a memoria, batería y CPU
 - ▶ Múltiples implementaciones de clientes MQTT disponibles para diferentes lenguajes
 - ▶ Ejemplo librerías en C de cliente en 30Kb y Java lib in 64Kb
- Diseñado para redes *constrained*:
 - ▶ El protocolo comprime las cabeceras y tiene campos variables para reducir tamaño
 - ▶ Menor tamaño posible de paquete: 2 bytes
 - ▶ Proporciona conciencia de sesión
 - ▶ Mensajes *keep alive* configurables para mantener sesión
 - ▶ Testado en diferentes tipos de redes VSAT, GPRS, 2G....
- Soporta Calidad de servicio QoS para asegurar la entrega de mensajes de forma determinista:
 - Niveles QoS:
 - ▶ 0 – mensaje enviado como mucho una vez (*fire and forget*) → entrega garantizada por TCP
 - ▶ 1 – mensaje entregado al menos una vez
 - ▶ 2 – mensaje entregado exactamente una vez
 - Productor y consumidor pueden tener niveles de QoS diferentes

MQTT

QoS downgrade



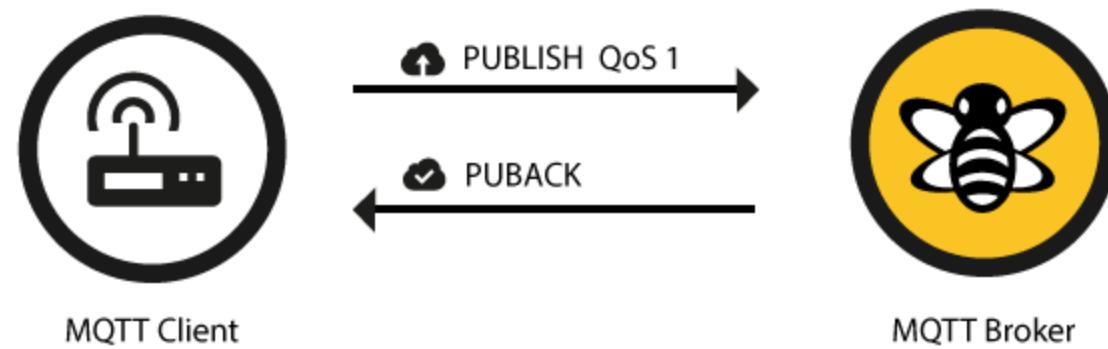
MQTT

QoS 0 – at most once



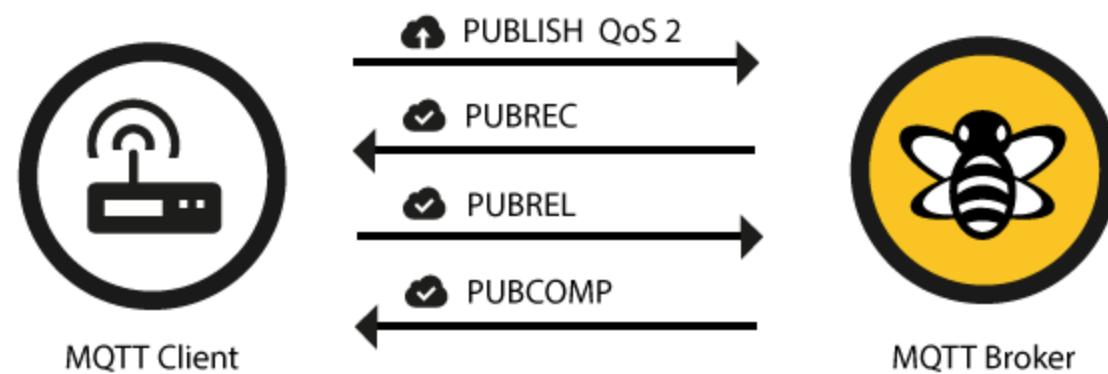
MQTT

QoS 1 – at least once



MQTT

QoS 2 – once and only once

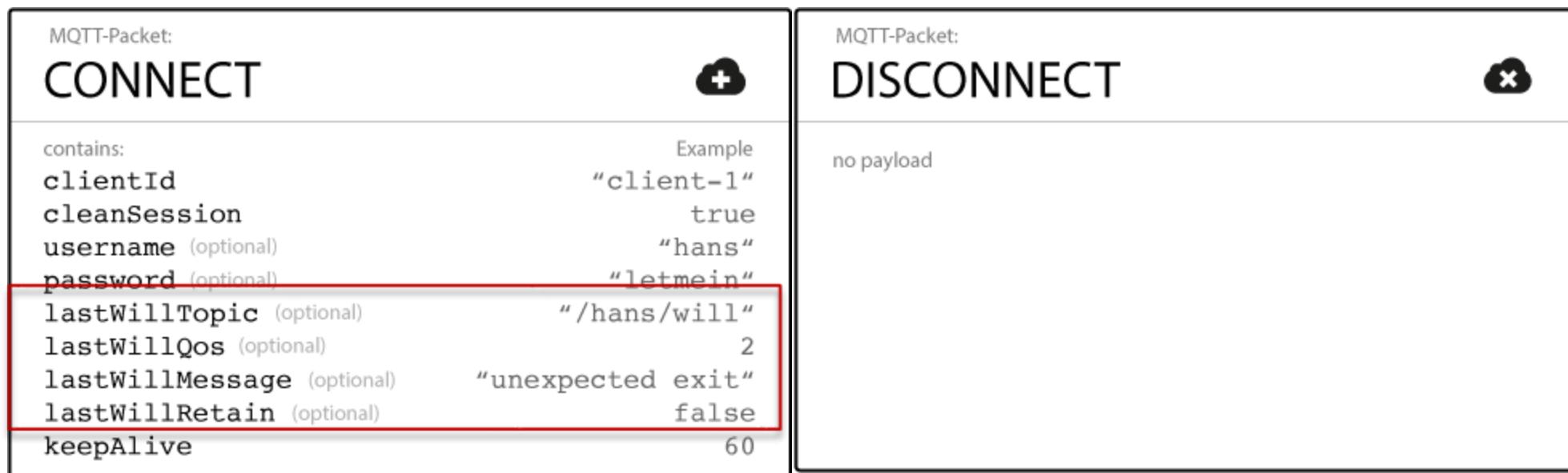


MQTT

- El *broker* es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal; los clientes mandan periódicamente un paquete (**PINGREQ**) y esperan la respuesta del broker (**PINGRESP**). *Keep alive* definido por el cliente en el *connect*. Max. *Keep alive* = 18h 12min 15 sec.
- La suscripción puede ser persistente o no-persistente (se define en el *connect*: *clean session*)
 - ▶ Persistente (*clean session* = *false*):
 - ▶ Una vez suscrito el broker envía los mensajes al suscriptor
 - Inmediatamente si el suscriptor está conectado
 - Si el suscriptor no está conectado los mensajes son almacenados en el *broker* hasta la próxima vez que se conecta el suscriptor.
 - ▶ No-persistente (*clean session* = *true*):
 - ▶ El tiempo de vida de la suscripción se limita al tiempo que el suscriptor está conectado al *broker*
 - ▶ Una paquete de publicación puede ser “*retained*”
 - ▶ El publicador marca el mensaje como *retained*
 - ▶ El *broker* guarda el ultimo mensaje marcado como *retained* para cada *topic*
 - ▶ El *broker* envía el ultimo mensaje *retained* a nuevos suscriptores cuando se conectan
 - ▶ Un nuevo suscriptor no tiene que esperar a que el publicador mande un nuevo mensaje, el *broker* directamente le manda el mensaje *retained*. Útil para no tener una “variable de estado” en blanco

MQTT

- *Last will and testament*
- Mensaje que el *broker* envía a todos los suscriptores cuando un productor se desconecta de forma inesperada (*PINGREQ out-of-time*)
- El mensaje se define en el *connect*, el *broker* lo almacena y sólo se envía si se detecta una desconexión sin que el productor haya enviado un mensaje *disconnect*



MQTT

▶ Tipos de paquetes de control

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	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

MQTT

▶ Ejemplo: PUBLISH message

▶ Fixed header:

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type (3)				DUP flag	QoS level		RETAIN
	0	0	1	1	X	X	X	X
byte 2	Remaining Length							

▶ En la variable header se incluye:

- ▶ Topic name and, opcionalmente, el identificador del paquete (in case QoS is used)

- ▶ El dato publicado se incluye en el payload

MQTT

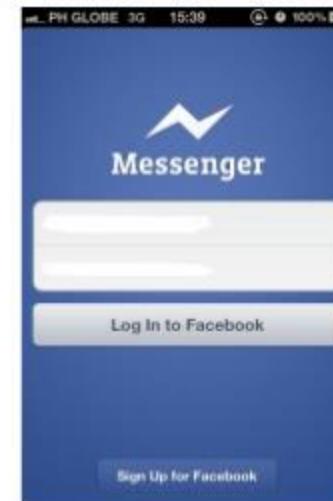
- ▶ MQTT-SN: *MQTT for Sensor Networks*
- ▶ Aunque diseñado para dispositivos muy limitados, MQTT aún puede ser demasiado pesado para ciertos casos específicos:
 - ▶ Mantener conexión TCP
 - ▶ *Topics* excesivamente largos para algunos protocolos de capas inferiores (ej. 802.15.4)
- ▶ MQTT-SN: para dispositivos embebidos, sobre UDP
- ▶ Rediseño de algunos mensajes, predefinición (indexado) de algunos *topics*

MQTT – Caso de uso

- ▶ *MQTT-based app: Facebook messenger*



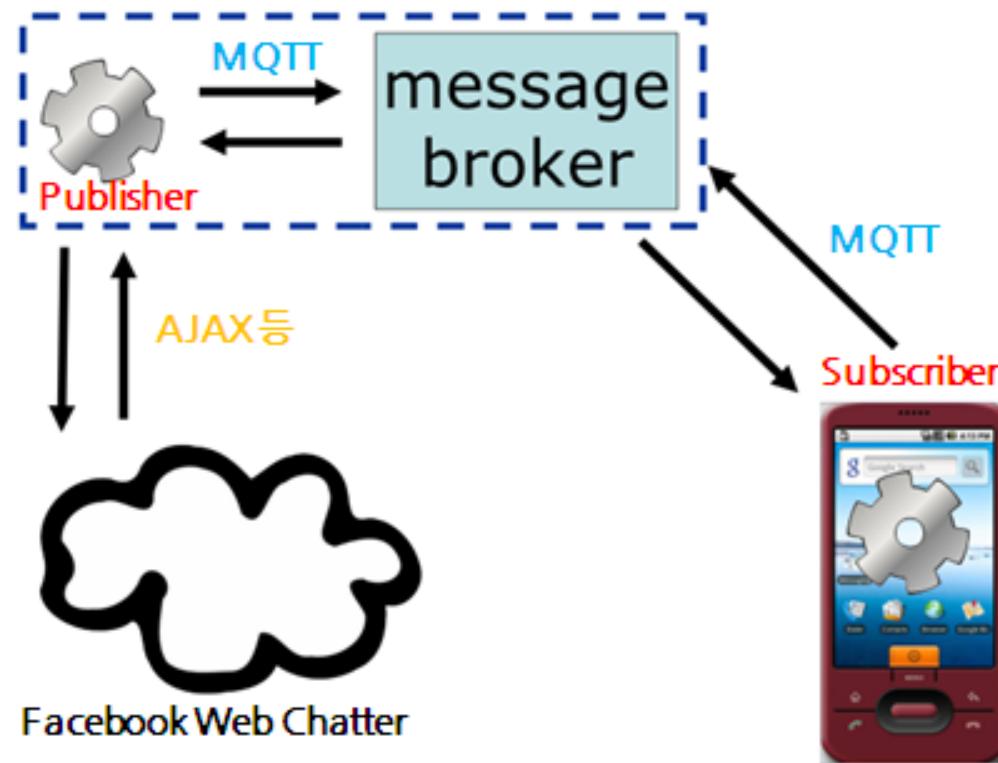
Mobile Software:
1. Facebook Messenger



- Facebook stated that they adopted MQTT to have faster phone to phone messaging while using **less battery and bandwidth**.
- MQTT can be used in iOS iPhone and iPad, Android, and Windows apps.

MQTT – Caso de uso

- ▶ MQTT-based app: Facebook messenger



Máster en Tecnologías de Análisis de Datos Masivos: BIG DATA

Internet de las Cosas en el Contexto de Big Data

PROTOCOLOS DE COMUNICACIONES EN IoT

Juan Antonio Martínez juanantonio@um.es
Luis Bernal Escobedo Luis.bernal@um.es

