

Internet de las Cosas

Máster Universitario en Inteligencia Computacional e
Internet de las Cosas.

Universidad de Córdoba

Bloque I: Visión técnica de IoT

Objetivos

- Conocer los protocolos y estándares más habituales, para las diferentes capas del modelo OSI, para desplegar soluciones IoT.
- Disponer de criterios efectivos para proponer soluciones válidas en cada caso.
- Adquirir una visión completa de lo que supone un despliegue IoT a nivel técnico.

Índice

10. Tecnología de las cosas

- a. UART
- b. SPI
- c. I2C
- d. OneWire
- e. ADC
- f. PWM

11. Protocolos a nivel Físico/Enlace

- a. El medio radioeléctrico
- b. 802.15.4
- c. Lora
- d. BLE
- e. UWB

12. Protocolos a nivel Transporte/Aplicación

- a. MQTT
- b. COAP

13. Bibliografía

Índice

10. Tecnología de las cosas

- a. UART
- b. SPI
- c. I2C
- d. OneWire
- e. ADC
- f. PWM

11. Protocolos a nivel Físico/Enlace

- a. El medio radioeléctrico
- b. 802.15.4
- c. Lora
- d. BLE
- e. UWB

12. Protocolos a nivel Transporte/Aplicación

- a. MQTT
- b. COAP
- c. ZigBee

13. Bibliografía

Protocolos a nivel Transporte/Aplicación

Nivel de transporte: garantizar comunicación correcta entre dos nodos conectados en red.

- Protocolos propios de este nivel: gestión de circuitos virtuales, reenvío de paquetes perdidos, reordenamiento de paquetes, etc.

Nivel de aplicación: el nivel de aplicación utiliza los niveles inferiores para implementar los servicios y programas cuando éstos requieren compartir datos con otros equipos de la red.

- Este es el nivel de usuario de la red (programador) y el él tiene cabida toda implementación imaginable.

Protocolos a nivel Transporte/Aplicación

Arquitecturas de distribución de carga

Una vez resuelta la conectividad, distribuir una aplicación entre equipos en red requiere organización

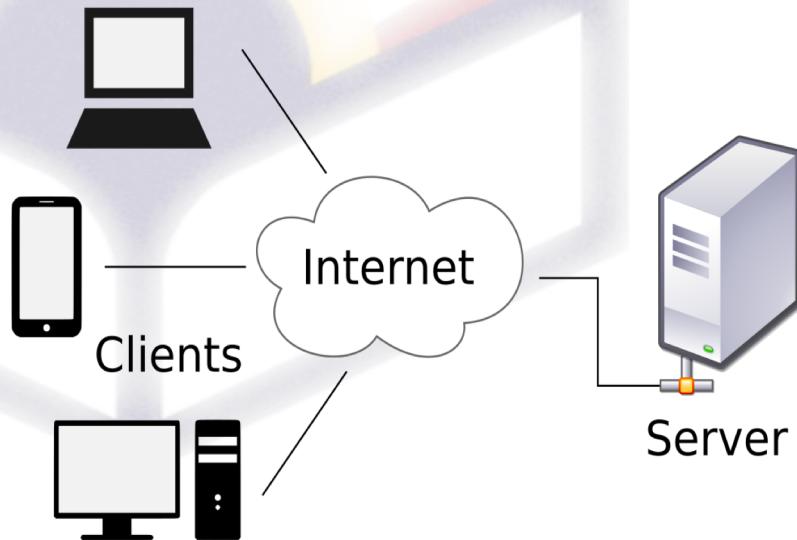
Considerando una tarea compartida, es importante determinar:

1. ¿Qué se envía? (modelo de datos)
2. ¿Dónde se almacena? (modelo de almacenamiento)
3. ¿Quién y a quién envía?
 - Roles:
 - Flujos de datos (cómo)
 - ¿Quién inicia la comunicación?
 - ¿Ciclos de trabajo? ¿eventos?

Protocolos a nivel Transporte/Aplicación

Arquitecturas de distribución de carga

I.- Arquitectura cliente-servidor

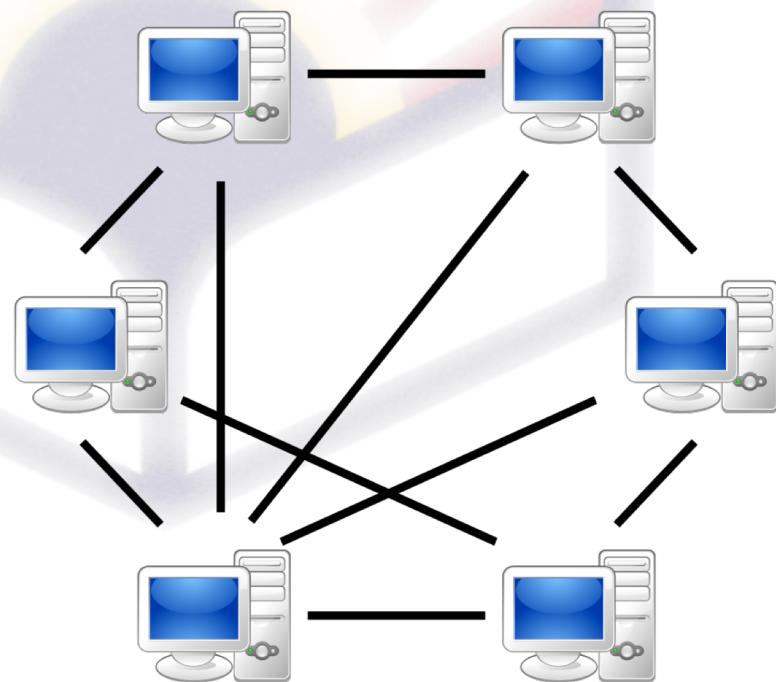


- Balance de carga y modelo de almacenamiento depende del servicio prestado.
- Equipos cliente
 - Ciclo de trabajo indeterminado
 - Procesan la invocación del servicio
- Equipos servidor
 - Permanentemente a la escucha (disponibilidad)
 - Diseño específico (hardware y software)
 - Servicio a múltiples clientes (dimensionamiento)
 - Cómputo
 - Recursos de red

Protocolos a nivel Transporte/Aplicación

Arquitecturas de distribución de carga

II.- Arquitectura P2P (Peer to Peer)

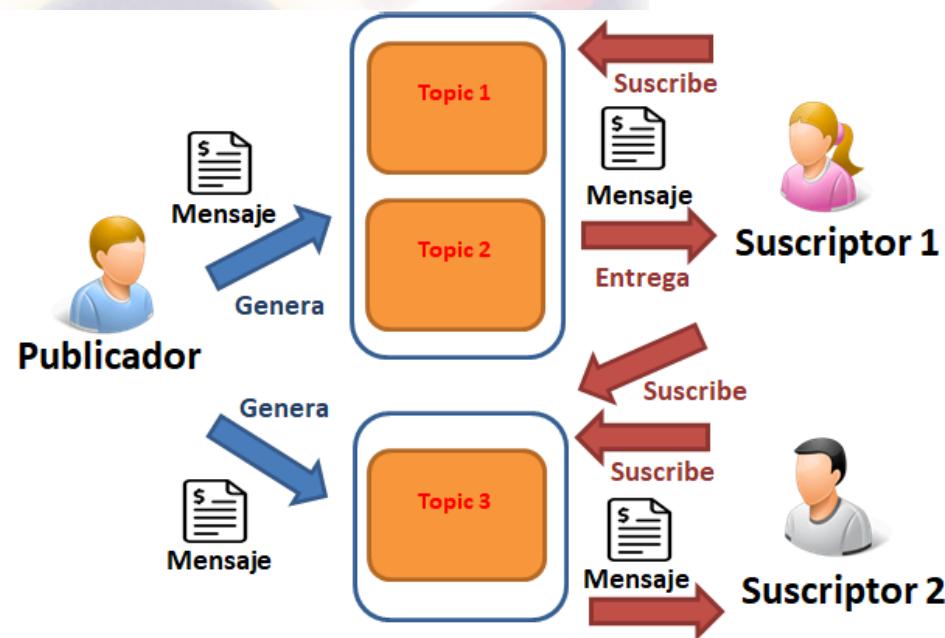


- Todos los equipos son clientes y servidores a ojos del resto
- Los “trabajos” se descentralizan y reparten entre 2 o más equipos.
 - Ejemplo: Blockchain
- Robustez
- Escalabilidad
- Descentralización
- Anonimato
- Seguridad
- Dependiendo de la aplicación puede ser necesario un servidor de “enganche” (ejemplo: Emule).

Protocolos a nivel Transporte/Aplicación

Arquitecturas de distribución de carga

III.- Arquitectura Publicador-Suscriptor



- Publicadores: generadores de datos
- Suscriptores: consumidores de datos
- La relación entre generadores y consumidores es dinámica, se gestiona mediante paquetes de inicio/fin de suscripción
- Intermediario según implementación
- Abstracción de la fuente de datos
- **Eficiencia**

Protocolos a nivel Transporte/Aplicación



Message Queue Telemetry Transport

1. Publicación / suscripción

Ejemplo: El dispositivo 2 recibirá los mensajes del dispositivo 1 si está suscrito al tópico que el dispositivo 1 publica. El bróker se encarga de esta difusión.



2. Tópicos

Los tópicos se representan por cadenas separadas por "/", de tal modo que éstas definen niveles jerárquicos.



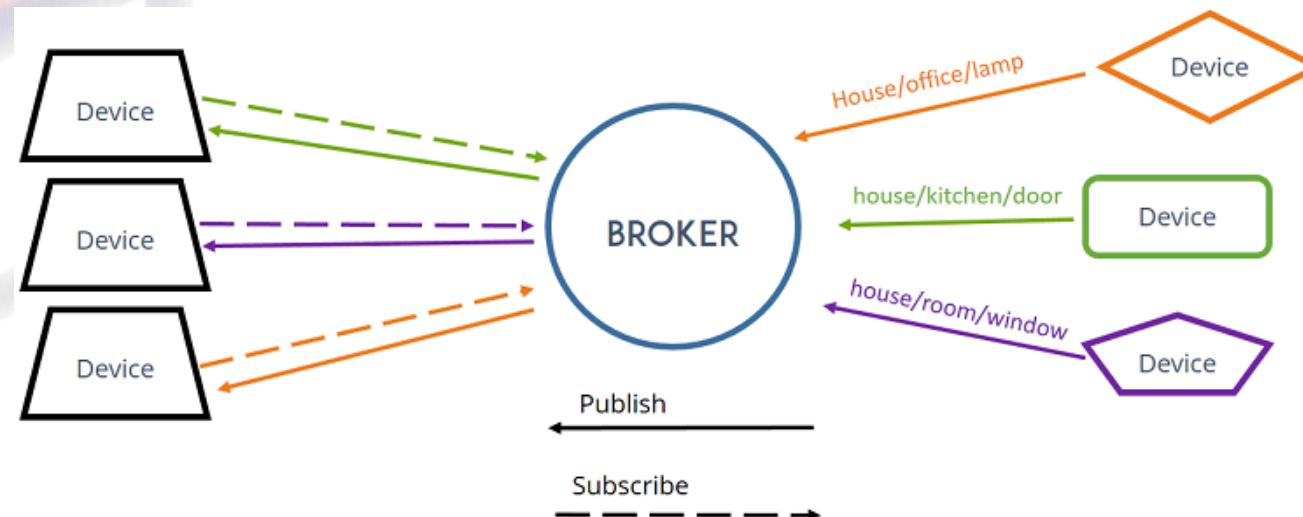
Protocolos a nivel Transporte/Aplicación



Message Queue Telemetry Transport

3. El broker

Existe un servidor central que recibe todos los mensajes, los filtra y los redirige a aquellos clientes interesados en la información.



4. Comandos

Cada mensaje de MQTT tiene asociado un comando, los más habituales son: CONNECT, DISCONNECT, SUBSCRIBE, UNSUBSCRIBE y PUBLISH.

Protocolos a nivel Transporte/Aplicación



Message Queue Telemetry Transport

5. Patrones de suscripción

Existen dos caracteres especiales para crear un patrón de suscripción:

- ‘+’ equivale a cualquier tópico en el nivel donde se encuentre el símbolo. Ejemplo: ‘/fábrica/+/temperatura’ recibirá publicaciones de ‘fábrica/control/temperatura’ y ‘fábrica/sala1/temperatura’.
- ‘#’ equivale a cualquier número de niveles a partir de la expresión previa. Solo se puede utilizar al final del patrón. Ejemplo: ‘fábrica/#’ recibirá publicaciones de “fábrica/control/temperatura” o “fábrica/sala1/humedad”

6. Control de la infraestructura

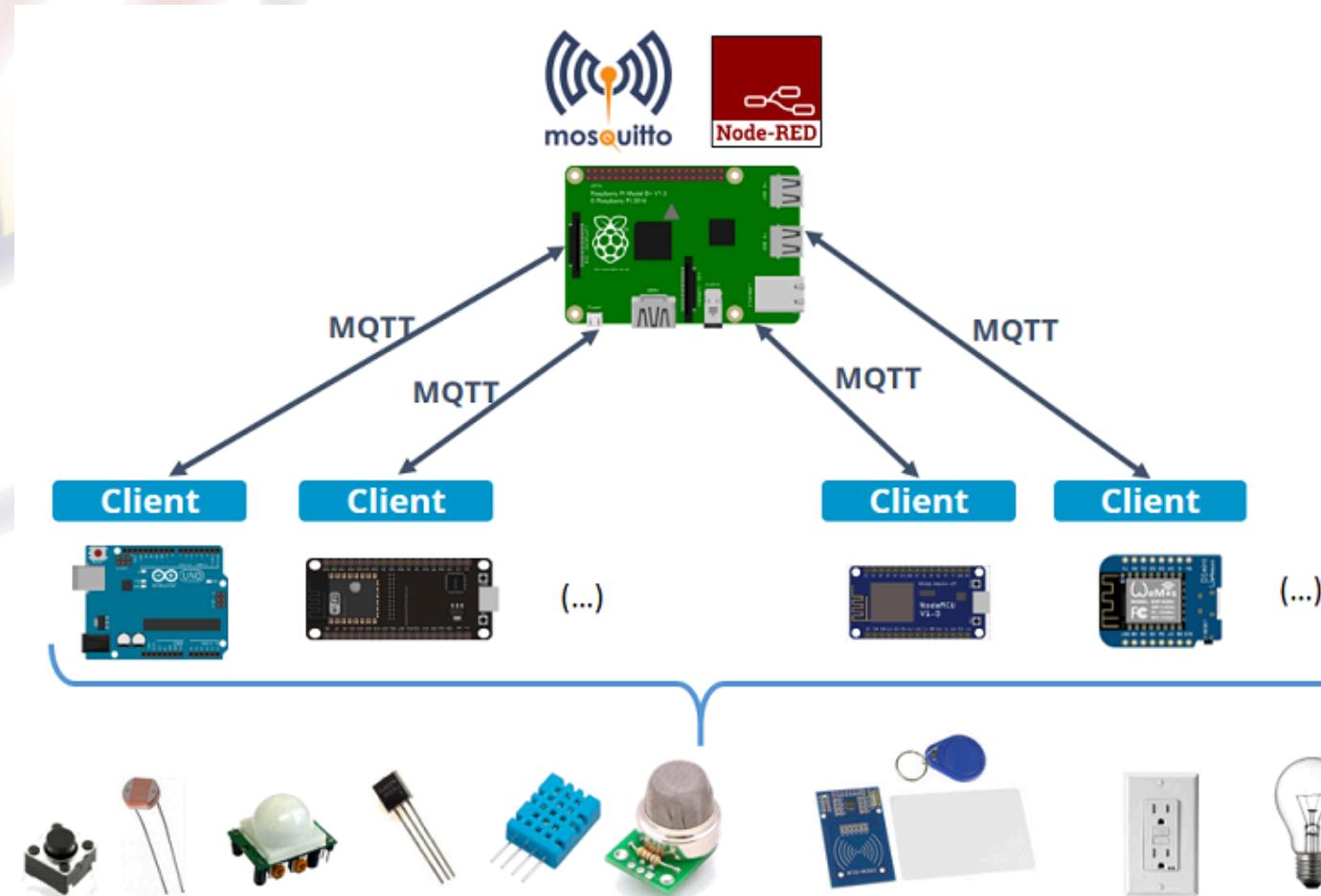
MQTT es un protocolo basado en TCP (orientado a conexión), mantiene las conexiones ‘abiertas’ y controla que siguen ‘vivas’, notificando ausencias a los nodos suscriptores si el publicador queda inoperativo.

7. Modos de calidad del servicio

MQTT define 3 modos de publicación:

- Nivel 0 (a lo sumo 1 vez): no confiable, no reenvío (confía en TCP)
- Nivel 1 (por lo menos 1 vez): reenvía si es necesario hasta obtener, al menos, 1 confirmación. Hay reenvíos y puede haber duplicados.
- Nivel 2 (exactamente 1 vez): se asegura de que el mensaje llega 1 sola vez (más seguro, más sobrecarga)

Protocolos a nivel Transporte/Aplicación

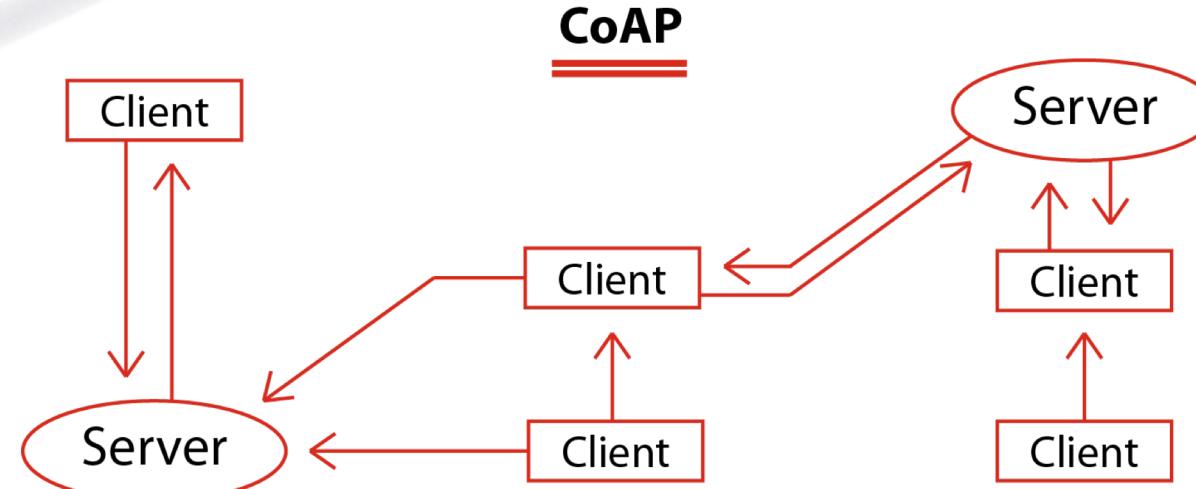


Protocolos a nivel Transporte/Aplicación

CoAP

Constrained Application Protocol

- Implementa la WEB de las cosas
- Basado en la arquitectura cliente-servidor (comunicación asíncrona, iniciada por el cliente)
- Diseñado para ser confiable incluso en escenarios de reducido ancho de banda y saturación.
- Similar a HTTP simplificado y comprimido, para las cosas.
- Seguridad mediante DTSL (Datagram Transport Security Layer) (UDP encriptado)



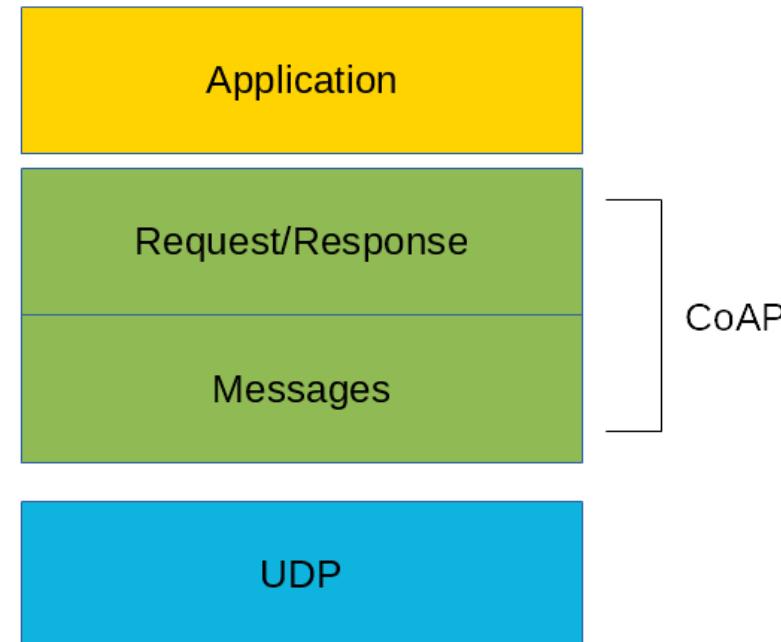
Protocolos a nivel Transporte/Aplicación

CoAP

Constrained Application Protocol

CoAP se subdivide en dos niveles de abstracción:

- Mensajes, la entidad más simple de información.
- Petición / Respuesta, una transacción entre cliente y servidor que controla el protocolo para que sea segura y confiable.



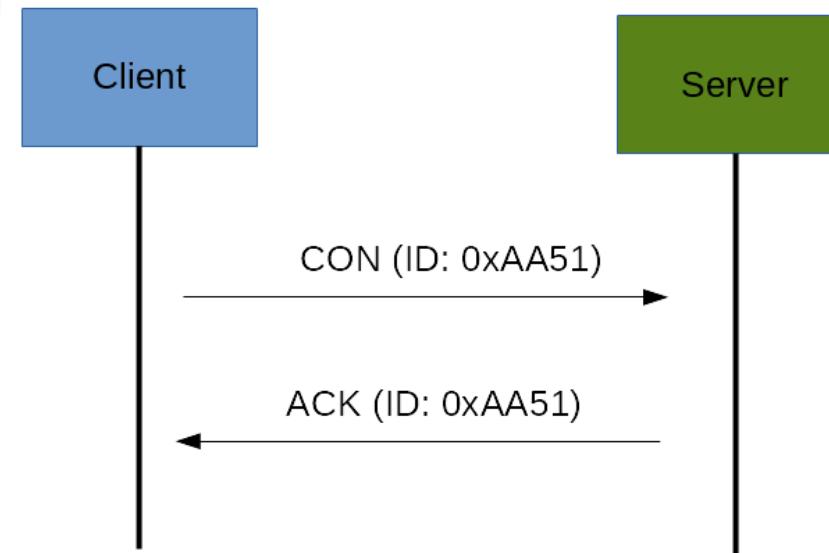
Protocolos a nivel Transporte/Aplicación

CoAP

Constrained Application Protocol

1. Mensajes

En CoAP existen dos tipos de mensajes: CONFIRMABLES y NO CONFIRMABLES. Los primeros implementan un protocolo confiable, los segundos no. Los mensajes tienen un ID único, lo que facilita la detección de duplicados. Siempre tienen un origen y un destino (ambos hosts de la red), si es confirmable, el destino responde con un datagrama de confirmación (ACK) con la misma ID del mensaje confirmado.



Protocolos a nivel Transporte/Aplicación

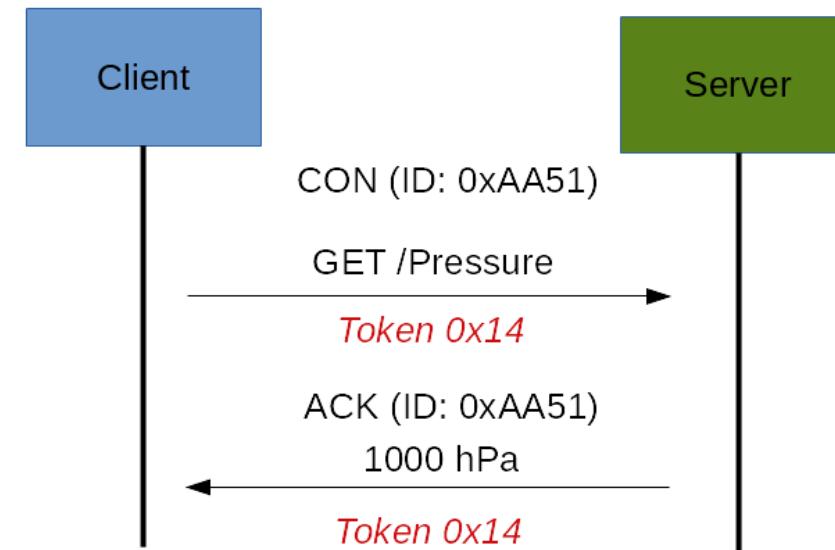
CoAP

Constrained Application Protocol

2. Petición / Respuesta (I)

A un nivel superior de abstracción, CoAP implementa un protocolo de petición / respuesta mediante mensajes. Si el servidor tiene algún problema con la petición y no puede facilitar respuesta a un mensaje confirmable, responde con un mensaje especial de RESET.

Si el servidor puede responder inmediatamente a una petición mediante mensaje confirmable, en el propio ACK se incluye la respuesta.



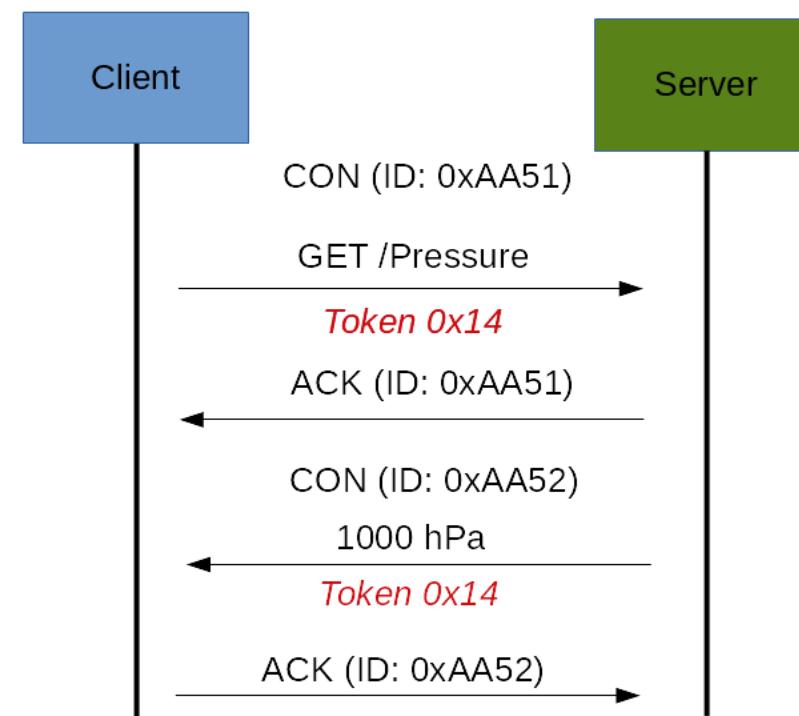
Protocolos a nivel Transporte/Aplicación

CoAP

Constrained Application Protocol

2. Petición / Respuesta (II)

Si por el contrario el servidor no puede responder inmediatamente, enviará un mensaje vacío al cliente y, tan pronto como le sea posible, le enviará la respuesta



Protocolos a nivel Transporte/Aplicación

CoAP

Constrained Application Protocol

2. Petición / Respuesta (III)

El mecanismo de petición / respuesta tiene su propio identificador, llamado Token, y se utiliza para casar peticiones y respuestas de manera segura.

3. Multicast

CoAP permite mensajes de uno a muchos, siempre no confiables.

4. Acceso a recursos API REST

CoAP implementa las 4 operaciones básicas para codificar con el modelo REST API (REpresentational State Transfer Application Program Interface): POST, GET, PUT, DELETE.

Ejemplo de URI CoAP

`coap://192.168.2.41/api/v1/$CREDENCIALES/parametros`
`coap://192.168.2.41/temperatura`

Protocolos a nivel Transporte/Aplicación



¿Preguntas?

Índice

10. Tecnología de las cosas

- a. UART
- b. SPI
- c. I2C
- d. OneWire
- e. ADC
- f. PWM

11. Protocolos a nivel Físico/Enlace

- a. El medio radioeléctrico
- b. 802.15.4
- c. Lora
- d. BLE
- e. UWB

12. Protocolos a nivel Transporte/Aplicación

- a. MQTT
- b. COAP
- c. ZigBee

13. Bibliografía

Bibliografía

Recursos bibliográficos para profundizar en los contenidos de la asignatura, todos ellos están en la librería digital de la Universidad de Córdoba

(<https://ebookcentral.proquest.com/lib/bibliocordoba-ebooks/home.action>):

Waher, Peter. *Learning Internet of Things*, Packt Publishing, Limited, 2015

Cirani, Simone, et al. *Internet of Things : Architectures, Protocols and Standards*, John Wiley & Sons, Incorporated, 2018

McEwen, Adrian, and Hakim Cassimally. *Designing the Internet of Things*, John Wiley & Sons, Incorporated, 2013

Internet of Things : Evolutions and Innovations, edited by Nasreddine Bouhaï, and Imad Saleh, John Wiley & Sons, Incorporated, 2017