ESPECIFICACIÓN PROTOCÓLO IMPLEMENTADO

Introducción

Este documento describe brevemente el protocolo CoAP implementado entre un servidor desarrollado en C y un cliente en Python.

Ambos se comunican mediante sockets UDP, siguiendo el modelo request/response e implementando los cuatro tipos de mensaje del protocolo junto con los métodos principales de operación.

Estructura del Mensaje

Cada mensaje CoAP incluye los campos: Version, Type, Token Length, Code, Message ID, Token, Options y Payload.

El payload transporta datos en formato JSON, normalmente relacionados con sensores (temperatura, humedad).

Tipos de Mensaje

CON (Confirmable): requiere confirmación; el servidor responde con un mensaje ACK.

NON (Non-confirmable): no requiere confirmación.

ACK (Acknowledgement): confirma la recepción de un CON.

RST (Reset): se usa cuando un mensaje no puede procesarse o es inválido.

Métodos Soportados

El servidor implementa los métodos GET, POST, PUT y DELETE, los cuales permiten la gestión completa de recursos:

GET: solicita el valor actual de un recurso.

POST: crea o registra nuevos datos.

PUT: actualiza valores existentes.

DELETE: elimina los datos del recurso.

Cada operación responde con un código CoAP según el resultado:

2.01 (Created), 2.04 (Changed), 2.05 (Content), 4.04 (Not Found) o 5.00 (Server Error).

Flujo de Interacción

El cliente envía solicitudes al servidor con un Message ID y un Token que permiten vincular respuestas y peticiones.

El servidor procesa la solicitud, actualiza o consulta los recursos en memoria y responde con el tipo y código adecuados.

Ejemplo típico:

POST /sensor → almacena datos.

GET /sensor → devuelve datos.

PUT /sensor → modifica un registro.

DELETE /sensor → elimina el recurso.

Logger y Almacenamiento

El servidor incluye un logger que muestra y guarda (en server.log) todas las solicitudes recibidas, incluyendo tipo de mensaje, cuerpo y respuesta enviada.

Los datos de sensores se gestionan en una tabla temporal en memoria (key/value).

Conclusión

La implementación cumple con la especificación del protocolo CoAP:

manejo de tipos de mensaje (CON, NON, ACK, RST), métodos CRUD, uso de tokens y Message ID, registro mediante logger, y operación sobre UDP con la API de Sockets Berkeley.