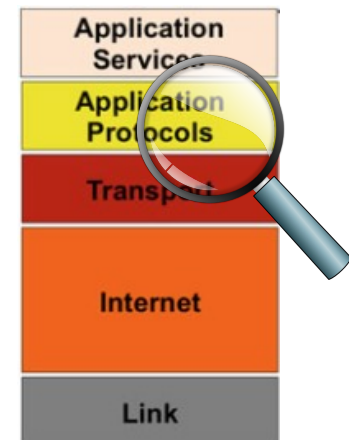


Ch. 13 - IoT Application Protocol Layer


Sec 2 – COAP Protocol

COMPSCI 147

Internet-of-Things; Software and Systems



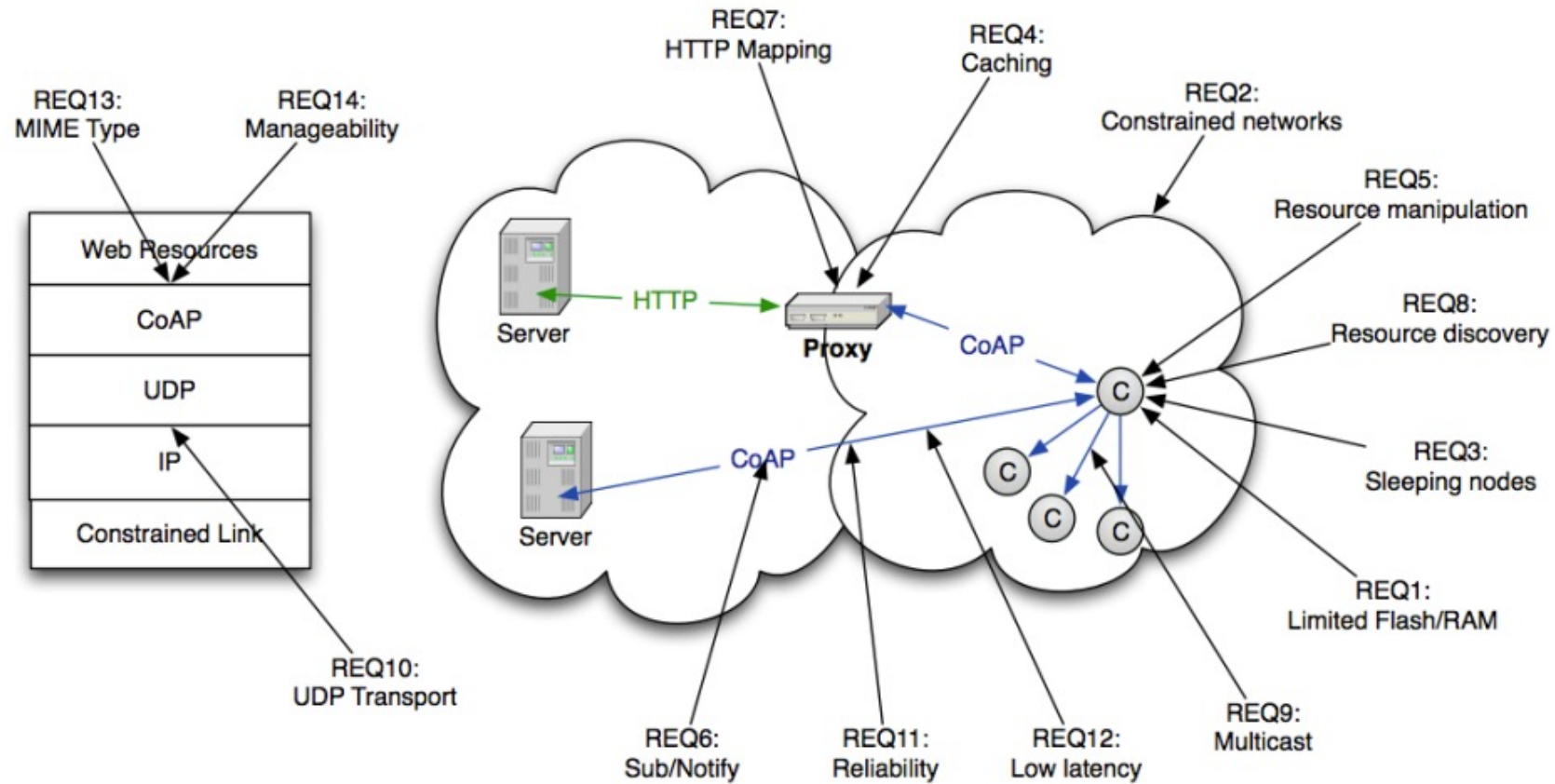
APP PROTOCOLS FOR IOT - STANDARDIZATION

- HTTP
 - IETF standard (RFC 2616 is HTTP/1.1)
-  • CoAP
 - IETF standard (RFC 7252)
- XMPP
 - IETF standard (RFC 6272)
- MQTT
 - OASIS standard
- AMQP
 - OASIS and ISO 19464 standard (1.0)
- SIP
 - IETF Standard (RFC 3261)
- IEEE 1888
 - IEEE Standard
- DDS (RTPS)
 - Object Management Group (OMG) Standard

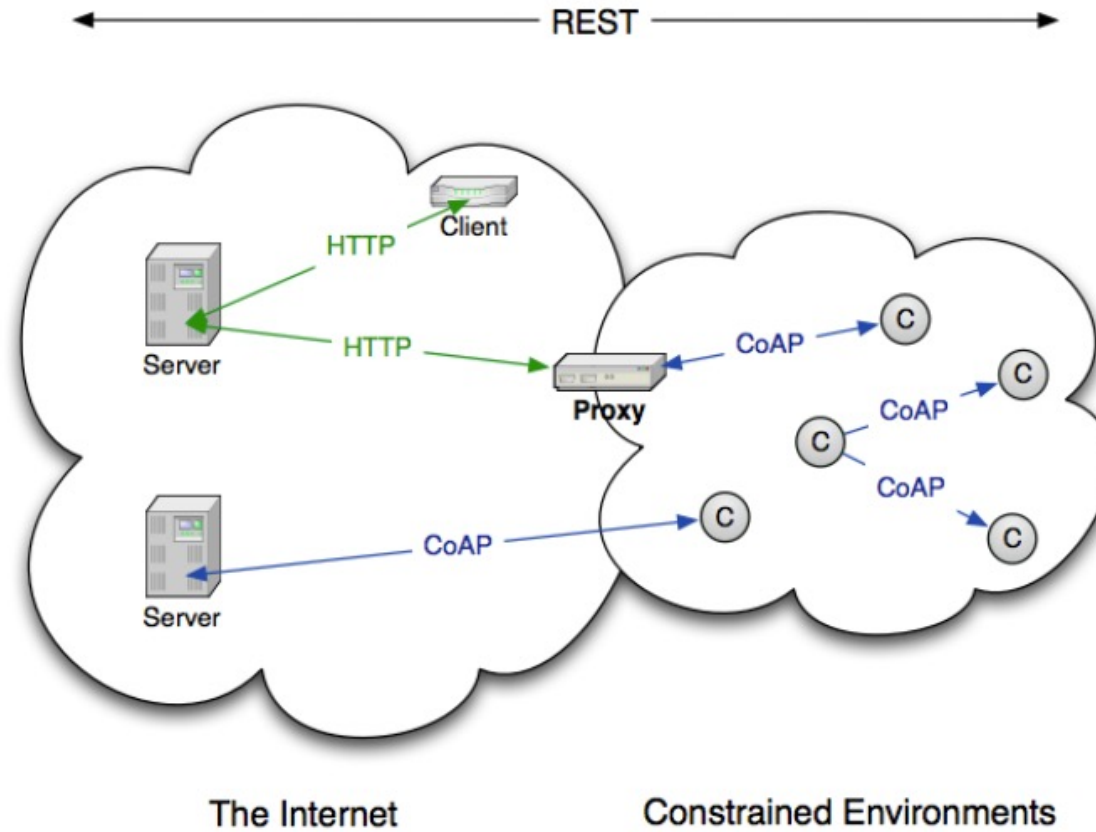
CORE AND COAP - CONSTRAINED RESTFUL ENVIRONMENTS

- IETF CORE: An **I**nternet **E**ngineering **T**ask **F**orce workgroup working on **C**onstrained **R**ESTful **E**nvironments to run on **I**P **n**etworks with **c**onstrained **r**esources.
- CoAP: Constrained Application Protocol is a specialized **I**nternet **A**pplication **P**rotocol for **c**onstrained **d**eVICES mainly by using **U**DP as transport and **s**hort **h**eaders.

CORE REQUIREMENTS



THE CORE ARCHITECTURE



COAP

- **Constrained** Application Protocol (CoAP)
 - Application-level protocol over UDP
 - Standardized by the IETF CORE workgroup as a **lightweight alternative** to HTTP
 - Powering M2M with a **Web of Things**
 - Designed to be used with **LLNs**



COAP II

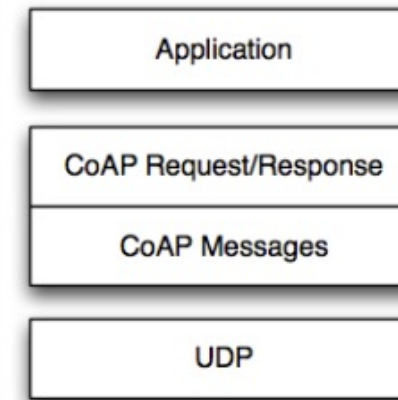
- **Embedded** web transfer protocol (coap://)
- Asynchronous transaction model
- UDP binding with reliability and multicast support
- GET, POST, PUT, DELETE methods
- URI support
- Small, simple header < 10 bytes
- HTTP-compatible response codes
- Optional observation, block transfer and discovery

WHAT COAP IS (AND IS NOT)

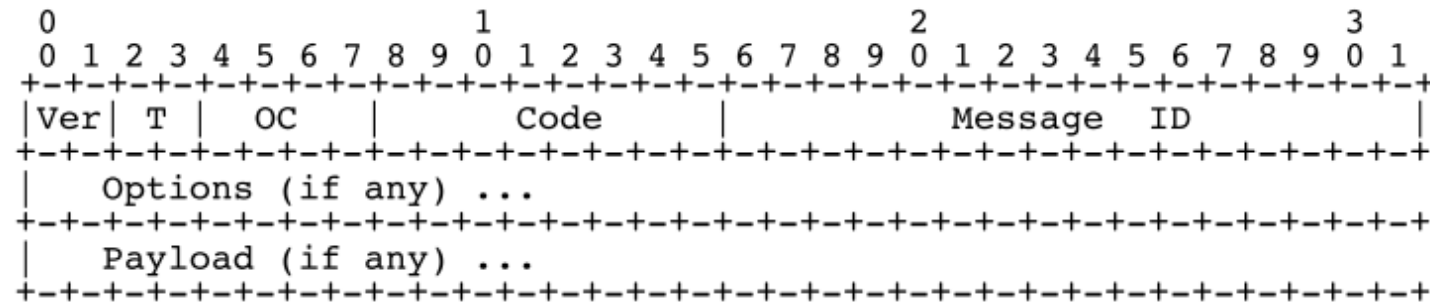
- CoAP is
 - A RESTful protocol
 - Both synchronous and asynchronous
 - For constrained devices and networks
 - Specialized for M2M applications
 - Easy to proxy to/from HTTP
- CoAP is not
 - A replacement for HTTP
 - General HTTP compression
 - Separate from the web

COAP - THE TRANSACTION MODEL

- Transport
 - CoAP is defined for UDP
- Messaging
 - Simple message exchange between end-points
 - CON – Confirmable
 - NON - Non-confirmable
 - ACK - Acknowledgment
 - RST - Reset message (not understood)
- REST
 - Request/Response piggybacked on messages



MESSAGE HEADER



Ver - Version (1)

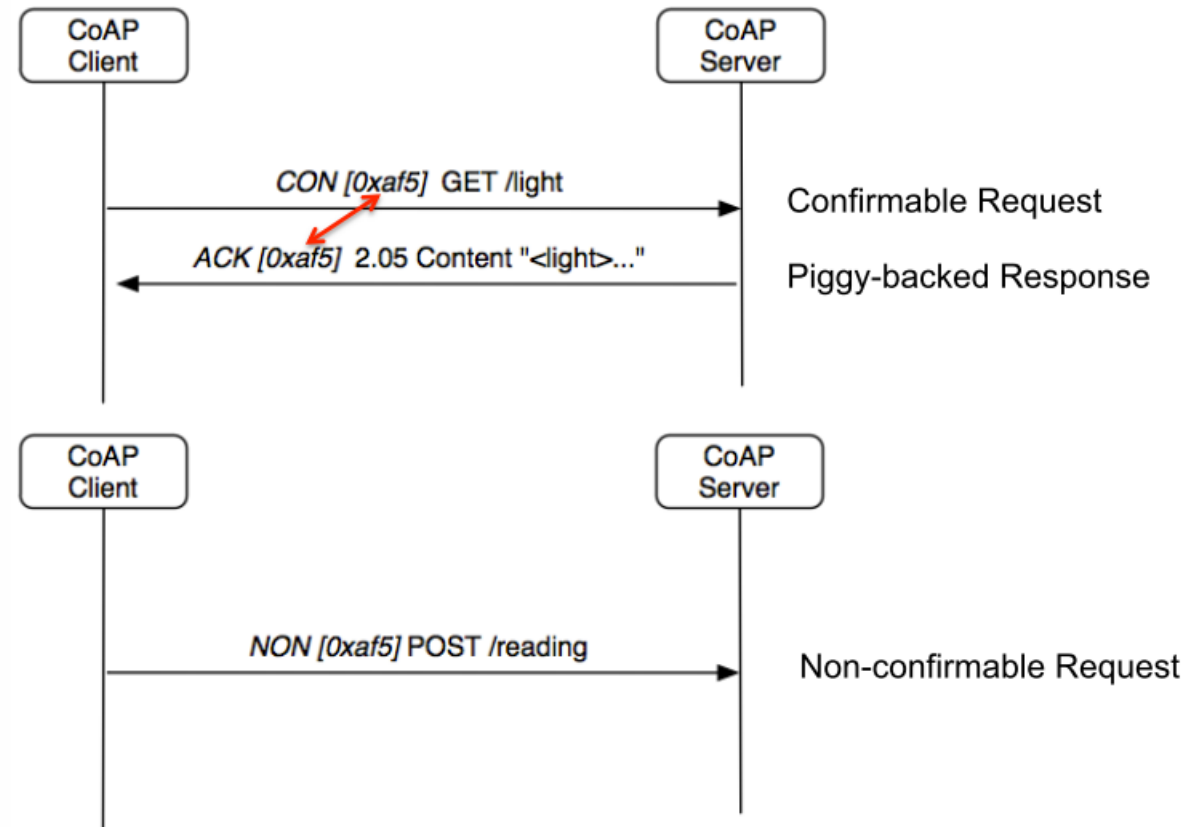
T - Transaction Type (Confirmable, Non-Confirmable, Acknowledgement, Reset)

OC - Option Count, number of options after this header

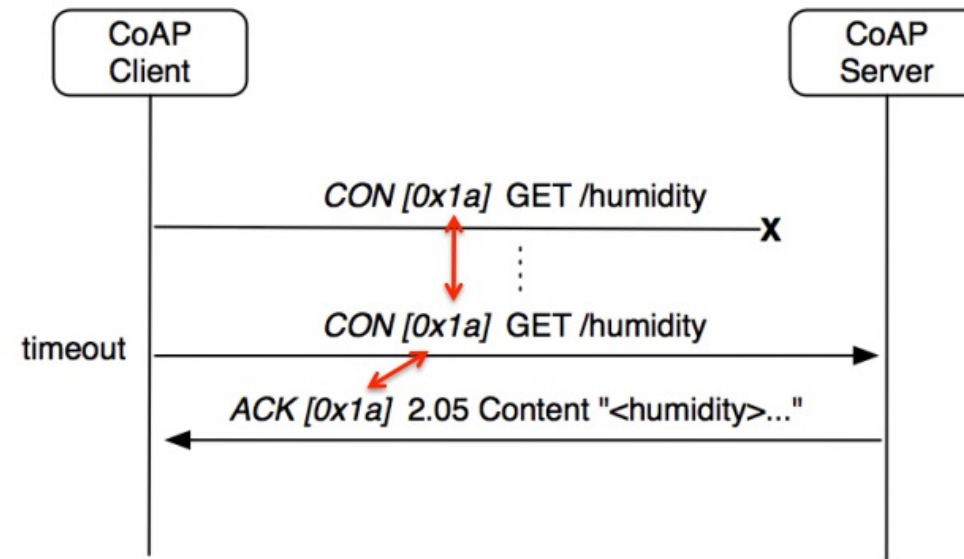
Code - Request Method (1-10) or Response Code (40-255)

Message ID - Identifier for matching responses

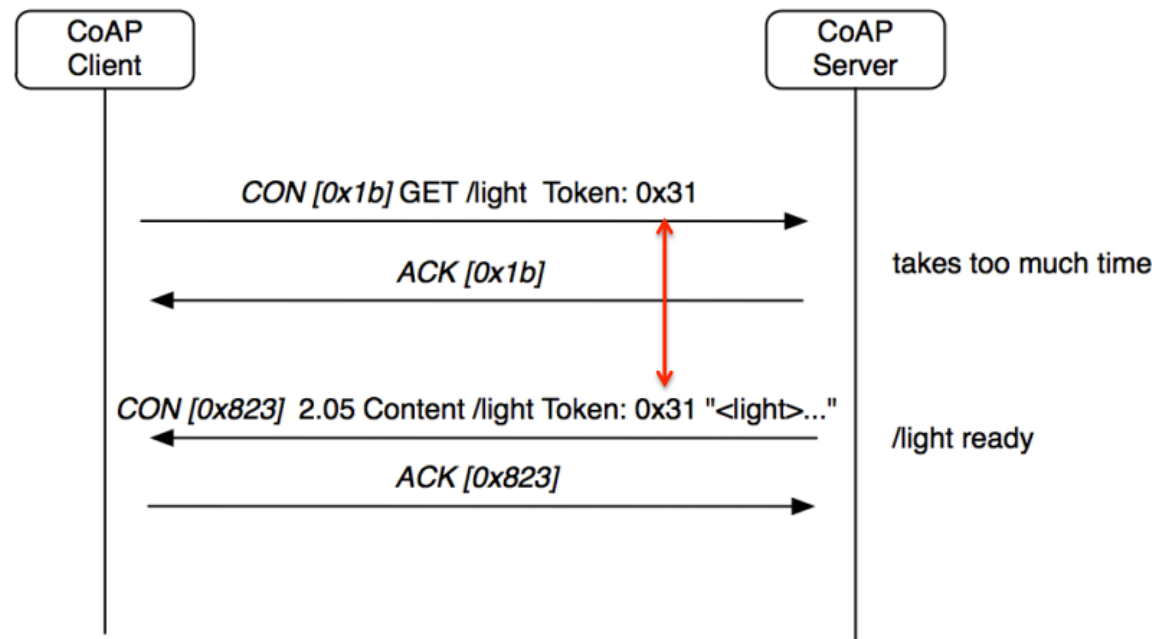
REQUEST EXAMPLES



DEALING WITH PACKET LOSS



NORMAL RESPONSE



The diagram illustrates a GET request and its acknowledgment (ACK) in a network protocol. It shows the structure of the packets, including sequence numbers, flags, and data fields.

GET Request Packet:

- Header:** CON [0x7d34] GET /temperature
- Sequence Number:** 0 (Client), 1 (Server)
- Flags:** GET = 1
- MID:** MID=0x7d34
- Data:** "temperature" (11 B) ...

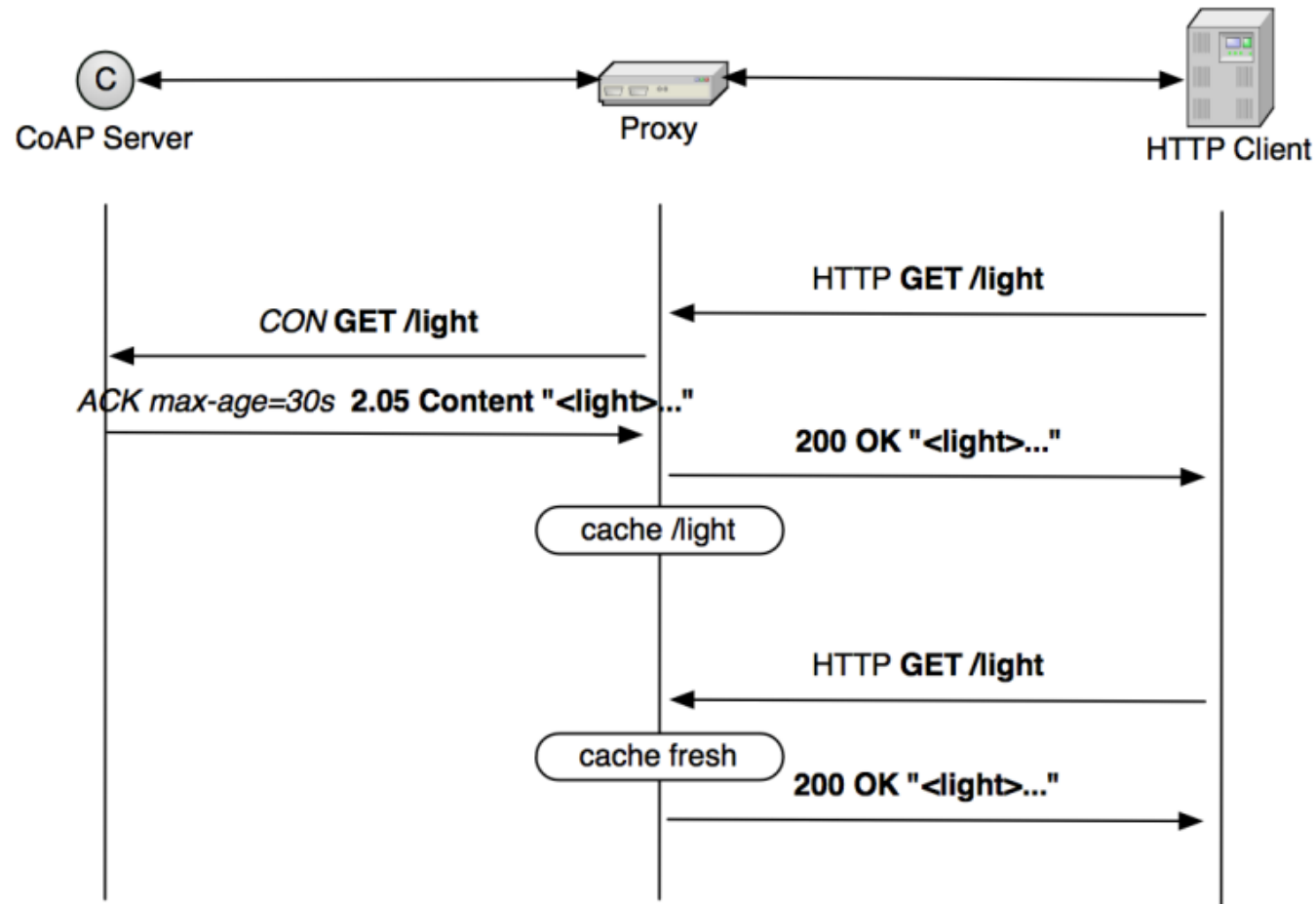
ACK Response Packet:

- Header:** ACK [0x7d34] 2.05 Content
- Sequence Number:** 0 (Client), 1 (Server)
- Flags:** 2.05=69
- MID:** MID=0x7d34
- Data:** "22.3 C" (6 B) ...

COAP - CACHING

- CoAP includes a simple caching model
 - Cacheability determined by response code
- Freshness model
 - **Max-Age** option indicates cache lifetime
- Validation model
 - Validity checked using the Etag Option
- A proxy often supports caching
 - Usually on behalf of **a sleeping node**,
 - and to reduce network load

PROXYING AND CACHING



Where is COAP used



What are you looking for?



Hej! Log in or sign up



Products Marketplace Rooms Design Deals

Costa Mesa 92697

Products > Lighting > Integrated lighting > Bookcase & cabinet lighting > TRÅDFRI Gateway

Top seller



TRÅDFRI

Gateway, white

★★★★☆ (66)

As low as \$7/mo for 6 months using the IKEA Projekt Credit Card**.

[Details >](#)



Works with IKEA Home Smart



Dimmable

Must be completed with



TRÅDFRI
Wireless dimmer
\$7.99



How to get it

How to run on ESP32

- <https://github.com/hirotakaster/CoAP-simple-library>
- Copper (Cu) CoAP user-agent browser extension

The screenshot displays the Copper (Cu) CoAP user-agent browser extension interface. The address bar shows the URL `coap://192.168.0.10:5683/LED`. The interface includes a toolbar with various CoAP methods (GET, POST, PUT, DELETE, Observe) and a sidebar for resource discovery. The main panel shows the CoAP message log, which contains two entries:

Time	CoAP Message	MID	Token	Options	Payload
16:39:22	CON-GET	30080 (0)	empty	Uri-Path: .well-known/core	
16:39:23	ACK-2.05 Content	30080	empty	Content-Format: 40	</LED>;LED;rt="observe";ct="0";title="observable resource"

A tooltip is visible over the 'LED' resource in the discovery tree, showing the following details:

- LED="true"
- rt="observe"
- ct="0"
- title="observable resource"

- Creating Smart Water Networks with Ayyeka
 - https://www.youtube.com/watch?v=w_eJL8XPtq0
- Constrained Application Protocol (CoAP) Tutorial
 - <https://www.youtube.com/watch?v=4bSr5x5gKvA>