

A Gentle Introduction to IoT Protocols: MQTT, CoAP, HTTP & WebSockets

Antonio Almeida and Jaime González-Arintero

June 14, 2017

Warning: we'll go fast!



Visionaries

>_ On-line Man-Computer Communication, 1962

113

ON-LINE MAN-COMPUTER COMMUNICATION

J. C. R. Licklider and Welden E. Clark

Bolt Beranek and Newman, Inc.

Cambridge, Massachusetts and Los Angeles, California

Summary

On-line man-computer communication requires much development before men and computers can work together effectively in formulative thinking and intuitive problem solving. This paper examines some of the directions in which advances can be made and describes on-going programs that seek to improve man-machine interaction in teaching and learning, in planning and design, and in visualizing the internal processes of computers. The paper concludes with a brief discussion of basic problems involved in improving man-computer communication.

ticularly adept.

For the kind of on-line man-computer interaction required in computer-centered military systems, a console featuring a Charactron display tube, a "light gun," and arrays of display lights and push buttons proved effective. At one time, about four years ago, at least 13 different companies were manufacturing such consoles -- different in minor respects but all alike in basic concept. Until recently, therefore, on-line man-computer communication could be summed up in the phrase: electric typewriters and SAGE consoles.

Visionaries

>_ The Computer as a Communication Device, 1968

The Computer as a Communication Device

In a few years, men will be able to communicate more effectively through a machine than face to face.

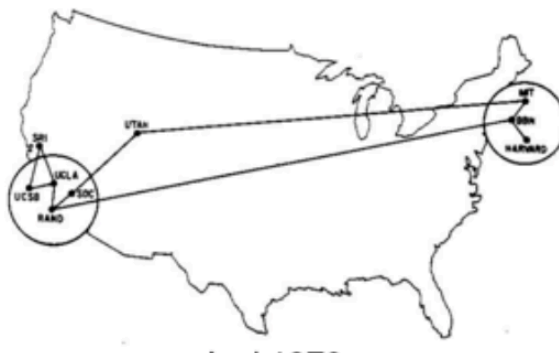
That is a rather startling thing to say, but it is our conclusion. As if in confirmation of it, we participated a few weeks ago in a technical meeting held through a computer. In two days, the group accomplished with the aid of a computer what normally might have taken a week.

We shall talk more about the mechanics of the meeting later; it is sufficient to note here that we were all in the same room. But for all the communicating we did directly across that room, we could have been thousands of miles apart and communicated just as effectively-as people-over the distance.

Arpanet



Dezember 1969



Juni 1970



März 1972



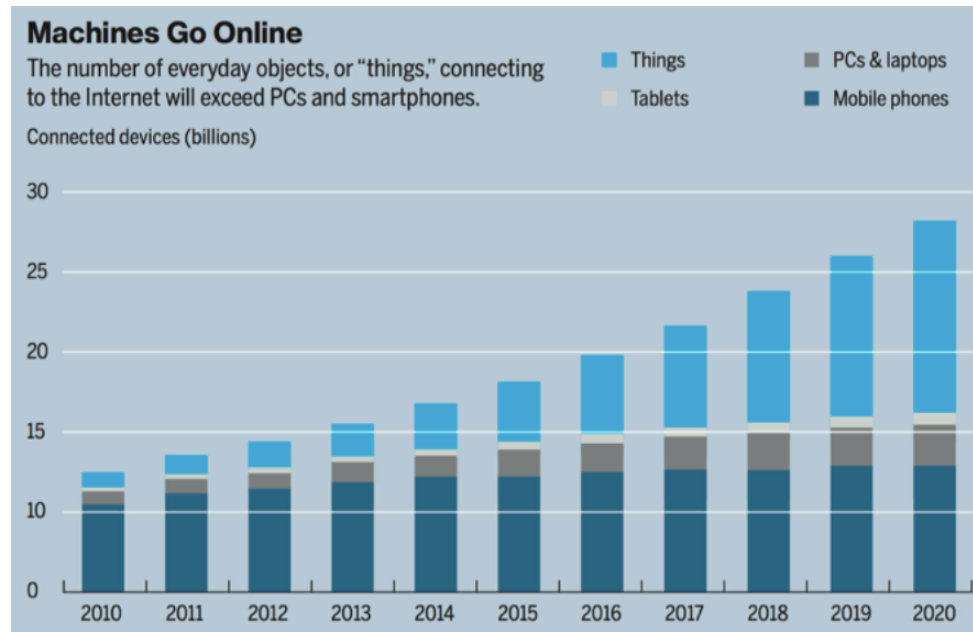
Juli 1977

Why more protocols?



Connected devices

>_ Some figures...



-MIT Technology Review, 2014

Connected devices

>_ Some figures...

- 14 bn connected devices | Bosch SI
- 50 bn connected devices | Cisco
- 309 bn IoT supplier revenue | Gartner
- 1.9 tn IoT economic value add | Gartner
- 7.1 tn IoT solutions revenue | IDC

*By 2020, component costs will have come down to the point that connectivity will become standard feature, even for processors costing **less than \$1**.*

- Peter Middleton, Gartner

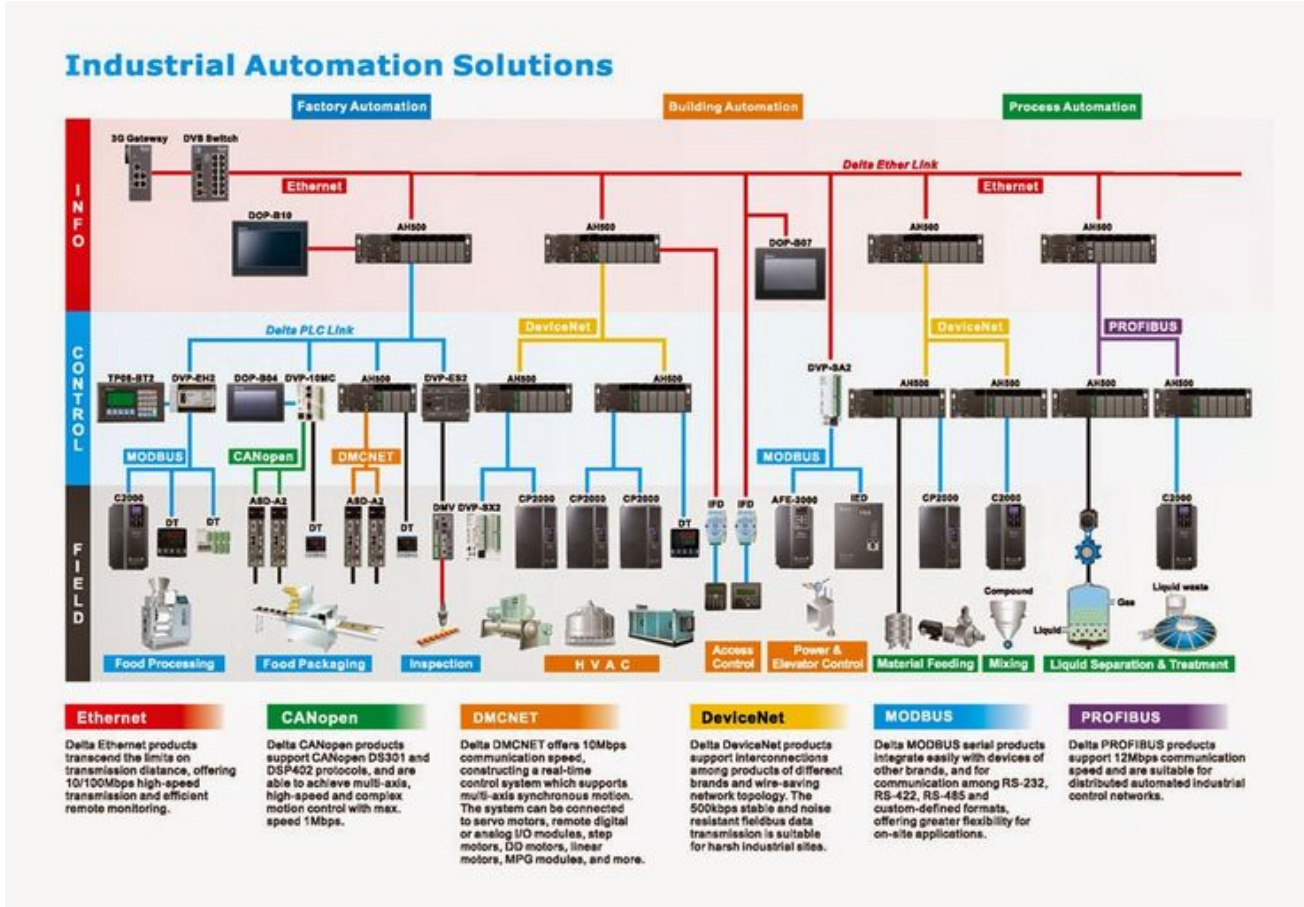
Constrained devices

- IETF Definition:
tools.ietf.org/html/rfc7228
- Limited processing power
- Unreliable networking
- Low power (*so they can run on batteries*)

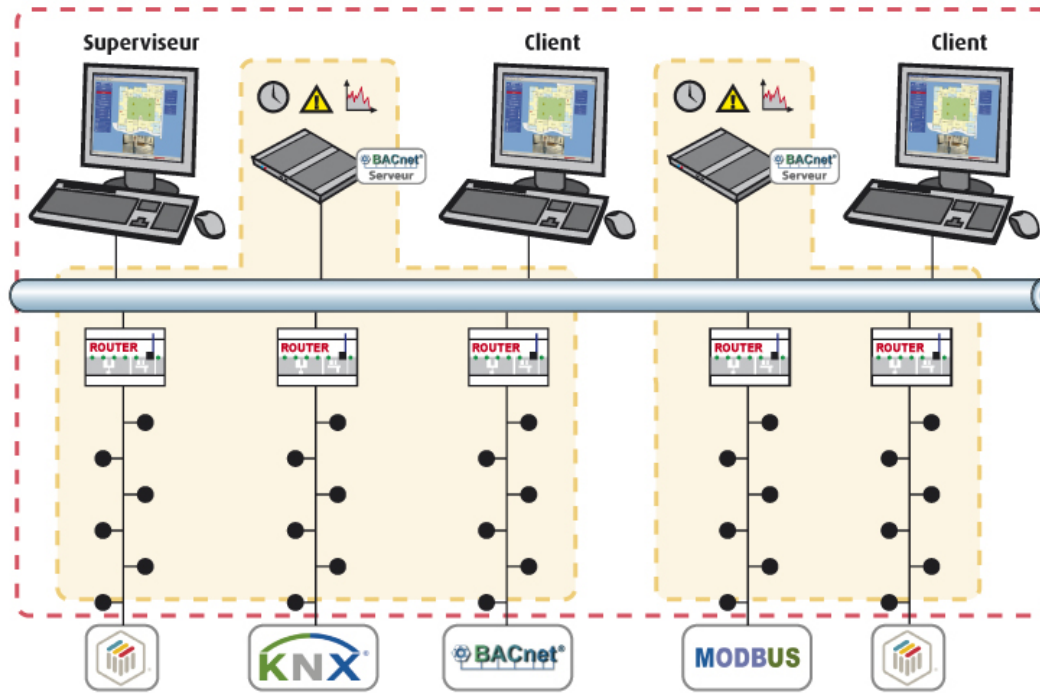
Internet: A definition

"A computer network consisting of a worldwide network of computer networks that use the TCP/IP network protocols to facilitate data transmission and exchange."

Several types of networks & protocols – industrial

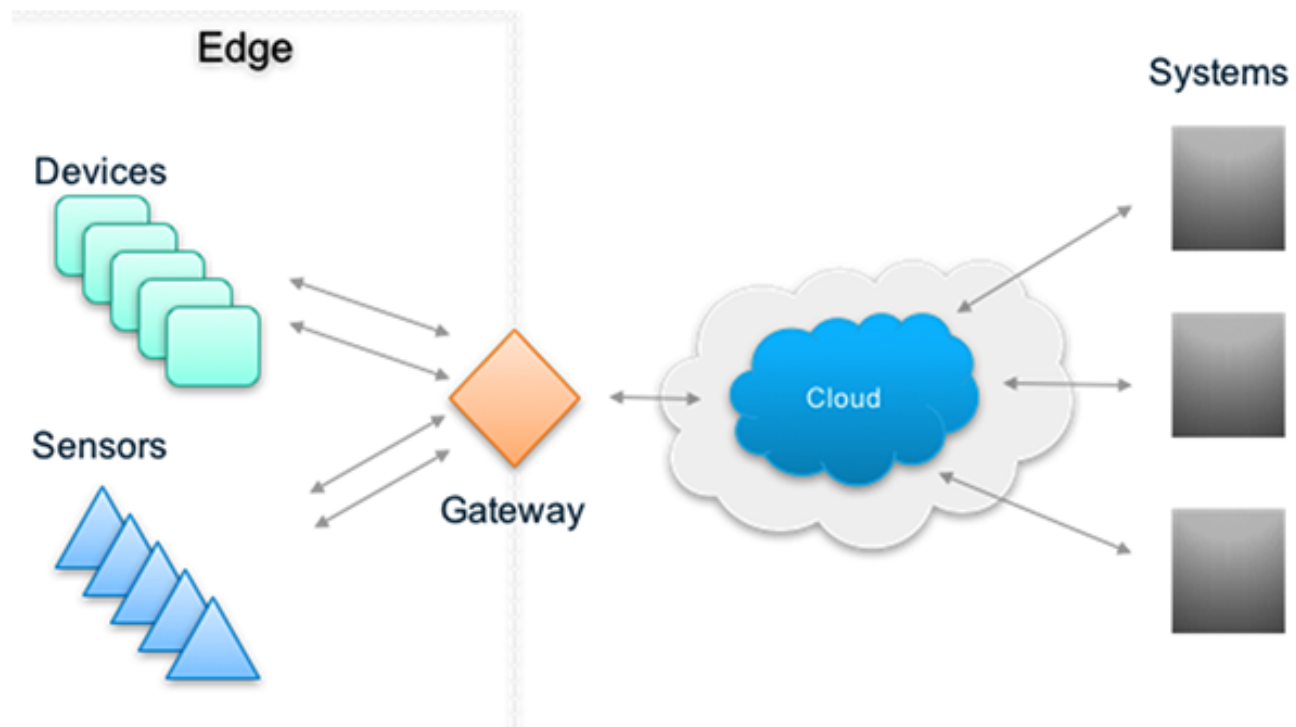


Other protocols – building management



Distributed architecture

***Edge* has devices — *Cloud* has servers**



Edge

devices != gateways

- Devices talk to other **devices northbound and southbound**
- Gateways talk to the **cloud northbound and devices southbound**
- Device to device (**D2D**)
- Device to cloud (**D2C**)

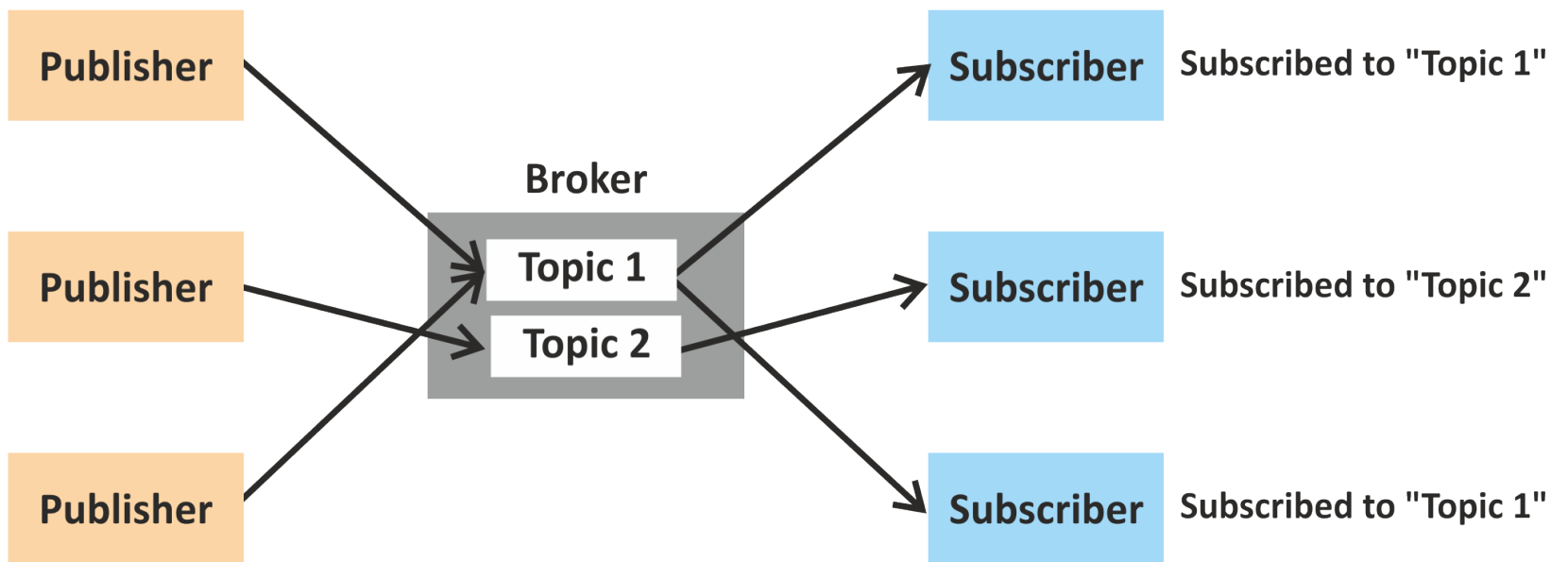
MQTT: Basics

Message Queue Telemetry Transport

"Publish-subscribe-based "lightweight" messaging protocol, for use on top of the TCP/IP protocol."

- Publish-subscribe
- A **message broker** is required
- Standard: ISO/IEC PRF 20922
- Small code footprint
- Limited network bandwidth / constrained environments
- Developed in 1999 (and released royalty free in 2010)
- Data agnostic

MQTT: Publish-subscribe model




MQTT: Connecting to the broker

Value	Return Code Response	Description
0	0x00 Connection Accepted	Connection accepted
1	0x01 Connection Refused, unacceptable protocol version	The Server does not support the level of the MQTT protocol requested by the Client
2	0x02 Connection Refused, identifier rejected	The Client identifier is correct UTF-8 but not allowed by the Server
3	0x03 Connection Refused, Server unavailable	The Network Connection has been made but the MQTT service is unavailable
4	0x04 Connection Refused, bad user name or password	The data in the user name or password is malformed
5	0x05 Connection Refused, not authorized	The Client is not authorized to connect
6-255		Reserved for future use

MQTT: Publishing to a topic

MQTT-Packet:

PUBLISH



contains:	Example
packetId (always 0 for qos 0)	4314
topicName	"topic/1"
qos	1
retainFlag	false
payload	"temperature:32.5"
dupFlag	false

MQTT: Subscribing to a topic

>_ Example topics

- Topic #1:
home/groundfloor/kitchen/temperature
- Topic #2:
office/conferenceroom/luminance

>_ Wild cards


- Single-level:
home/groundfloor+/temperature
*(to subscribe to **all the temperature readings** in all the rooms of the ground floor)*
- Multi-level: home/groundfloor/#
*(to subscribe to **all the readings** in all the rooms of the ground floor, **not only the temperature**)*

MQTT: Quality of Service

> _ QoS can be 0, 1, or 2

- 0: The broker/client will deliver the message once, with no confirmation.
- 1: The broker/client will deliver the message at least once, with confirmation required.
- 2: The broker/client will deliver the message exactly once by using a four step handshake.

MQTT: Last will and testament

MQTT-Packet:	
CONNECT	
	
contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
keepAlive	60

MQTT: Learn more

There are client libraries and wrappers for practically all languages used in M2M setups, as well as different brokers/servers.

- Learn more: mqtt.org
- Software: mqtt.org/software
- Recommended broker (C): **Mosquitto** (mosquitto.org)
- Lots of good tutorials out there

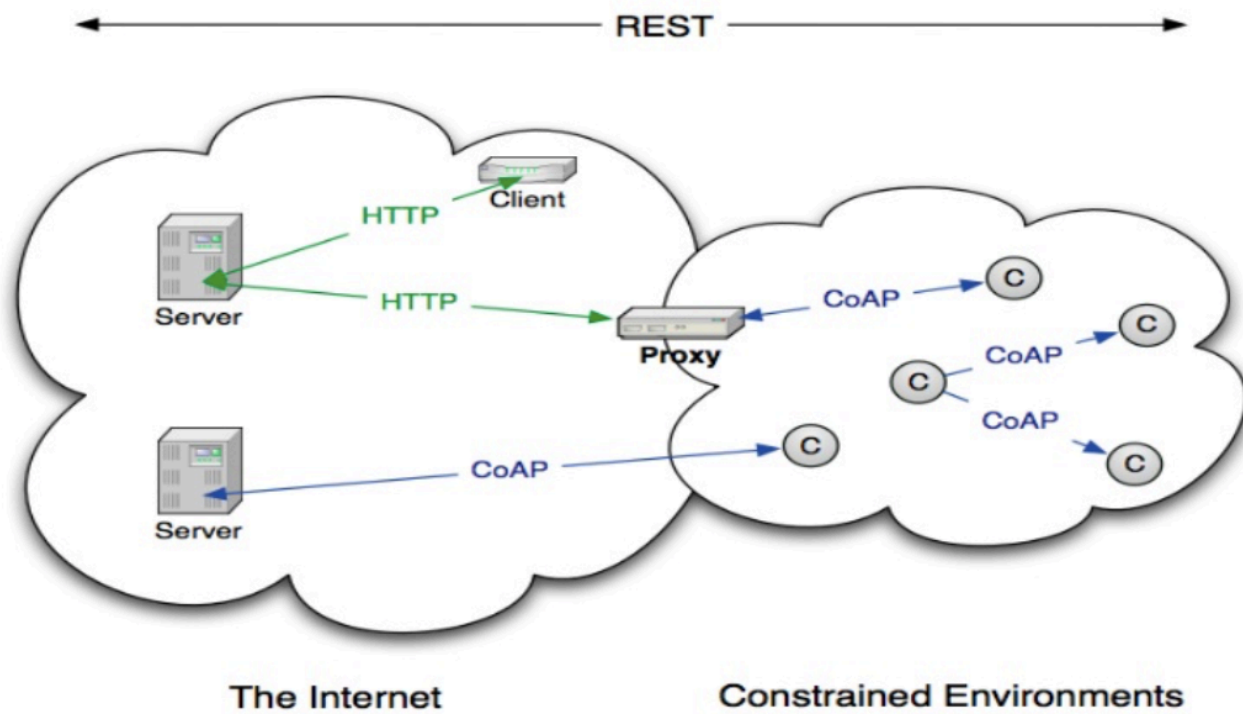
CoAP: Basics

Constrained Application Protocol

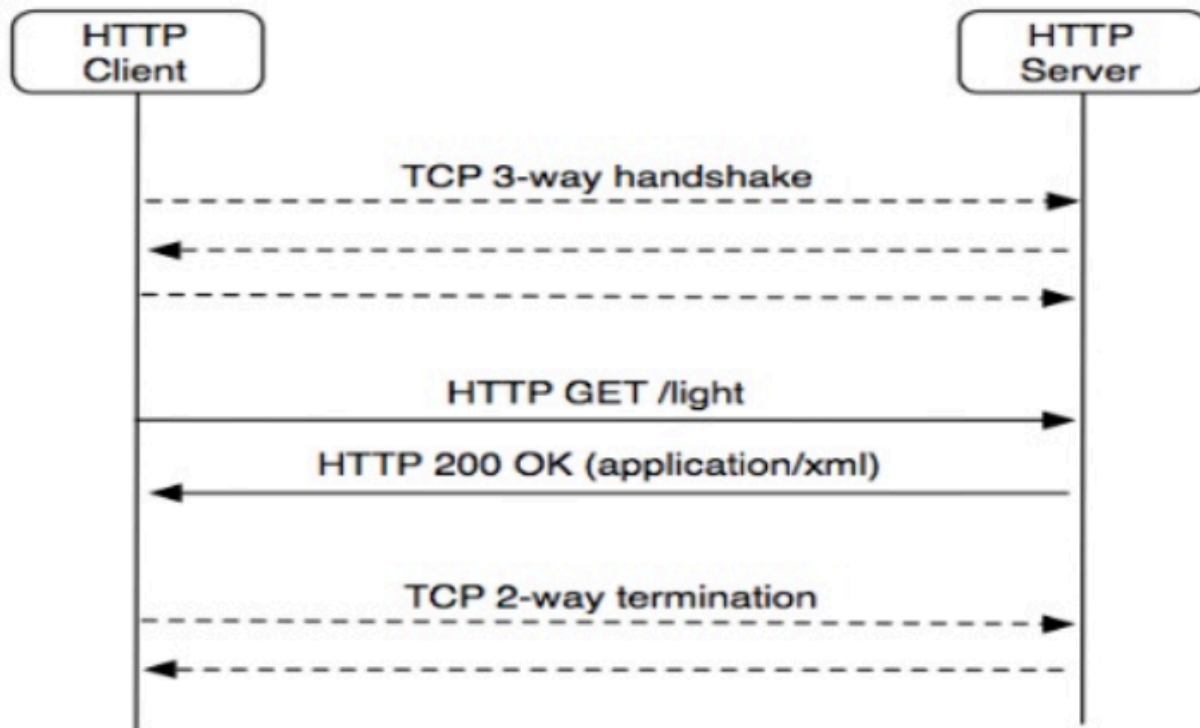
"A specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things."

- CoRE, IETF group
- Proposed standard: RFC 7252
- **CoAP ~ lightweight fast HTTP**
- Designed for manipulation of simple resources on constrained node networks

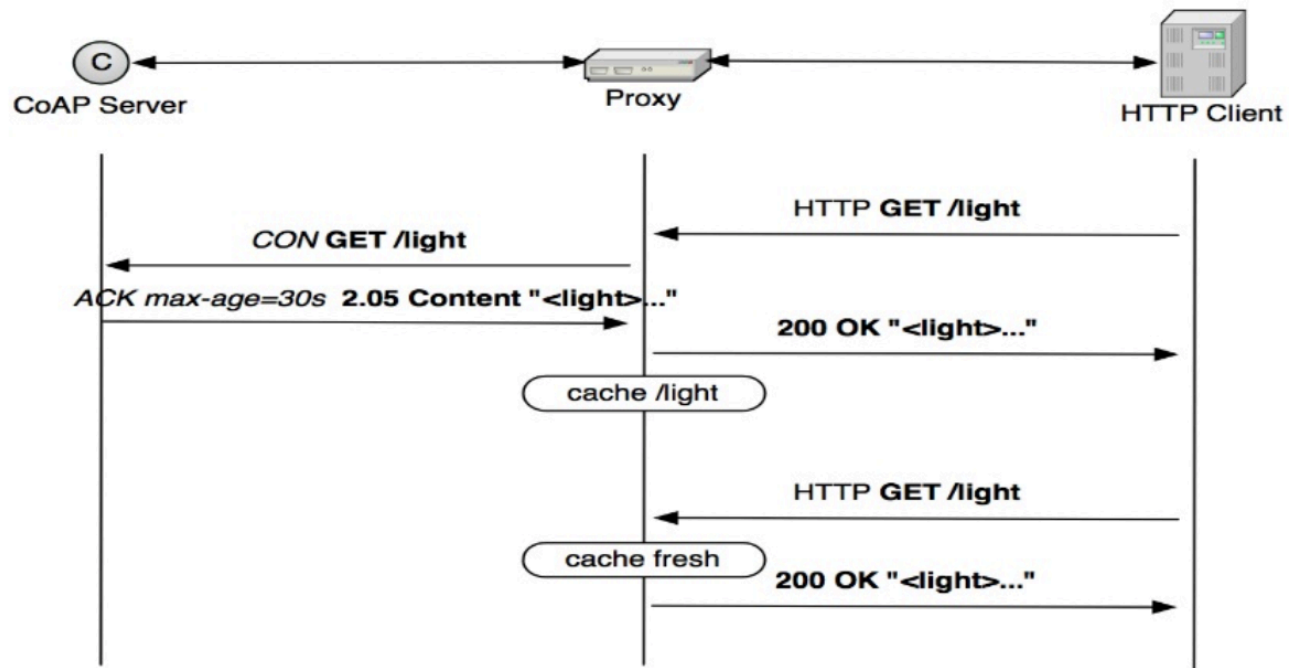
CoAP: RESTful environment



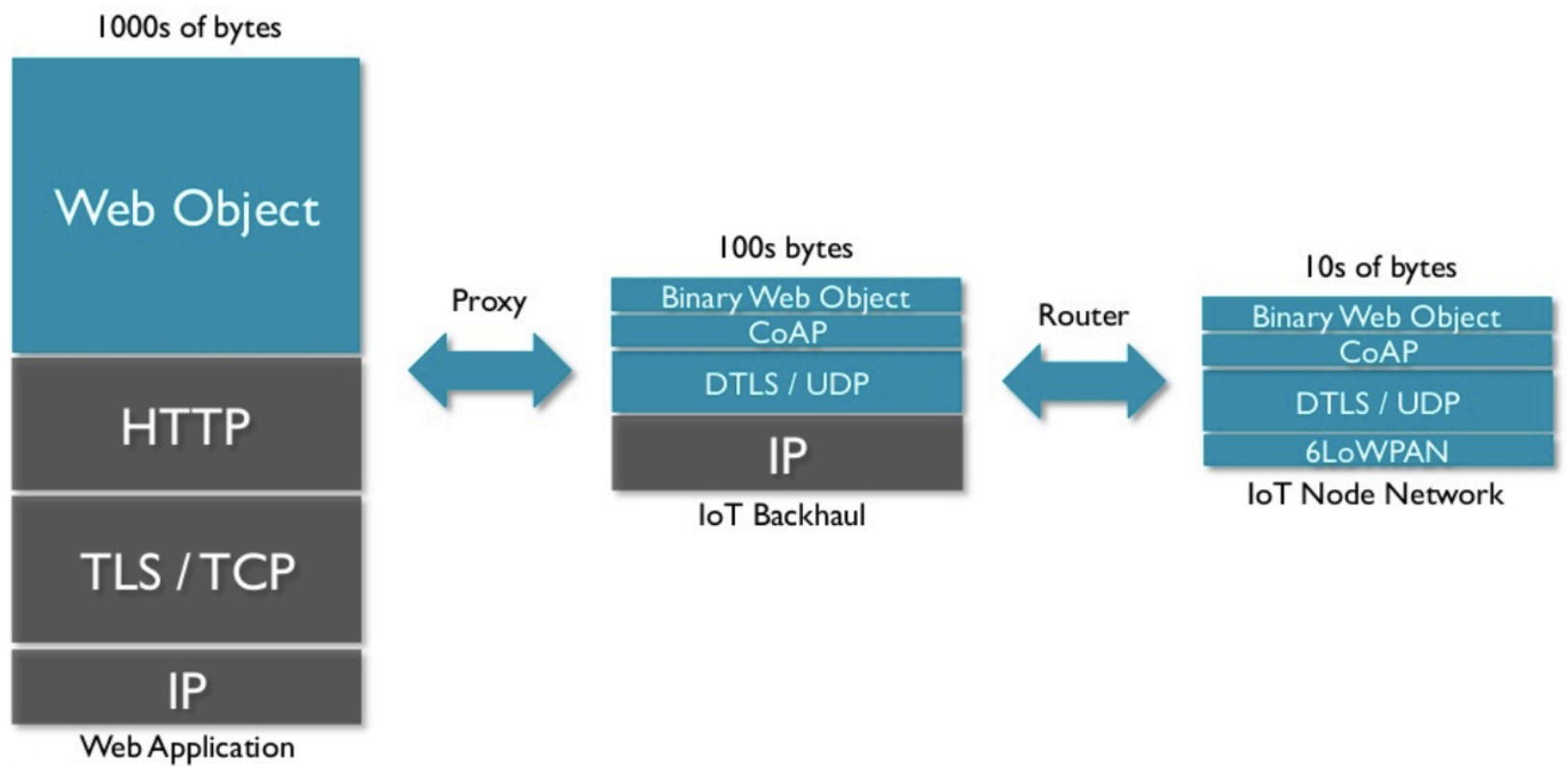
CoAP: Inspired by HTTP



CoAP <---> Proxy <---> HTTP



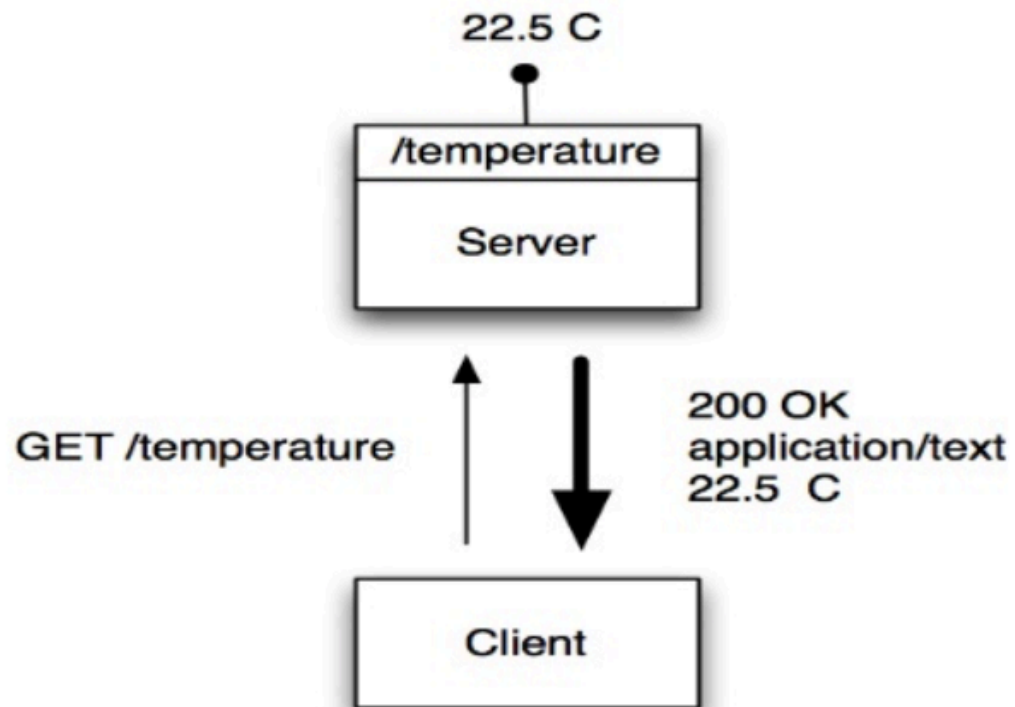
CoAP: HTTP vs CoAP



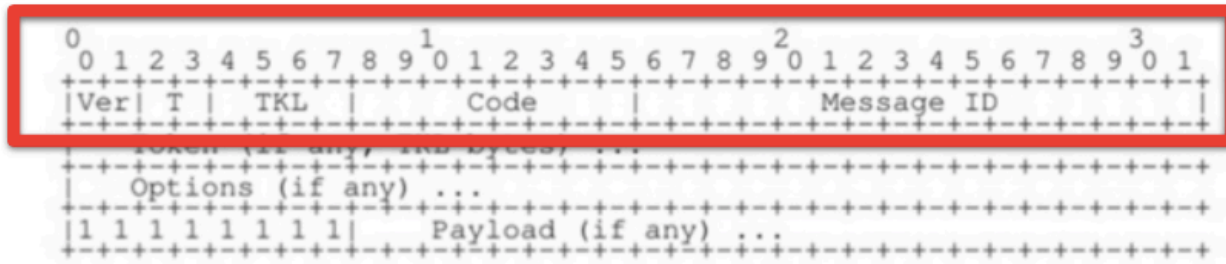
CoAP: Functionalities

- URI
- GET / POST / PUT / DELETE
- Content-type support (XML, JSON, CBOR,...)
- **Built-in discovery** — `.well-known/core`
- **Multicast support**
- Asynchronous message exchanges
- Designed to be extensible

CoAP: Client / Server



CoAP: Message format



Ver - Version (1)

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

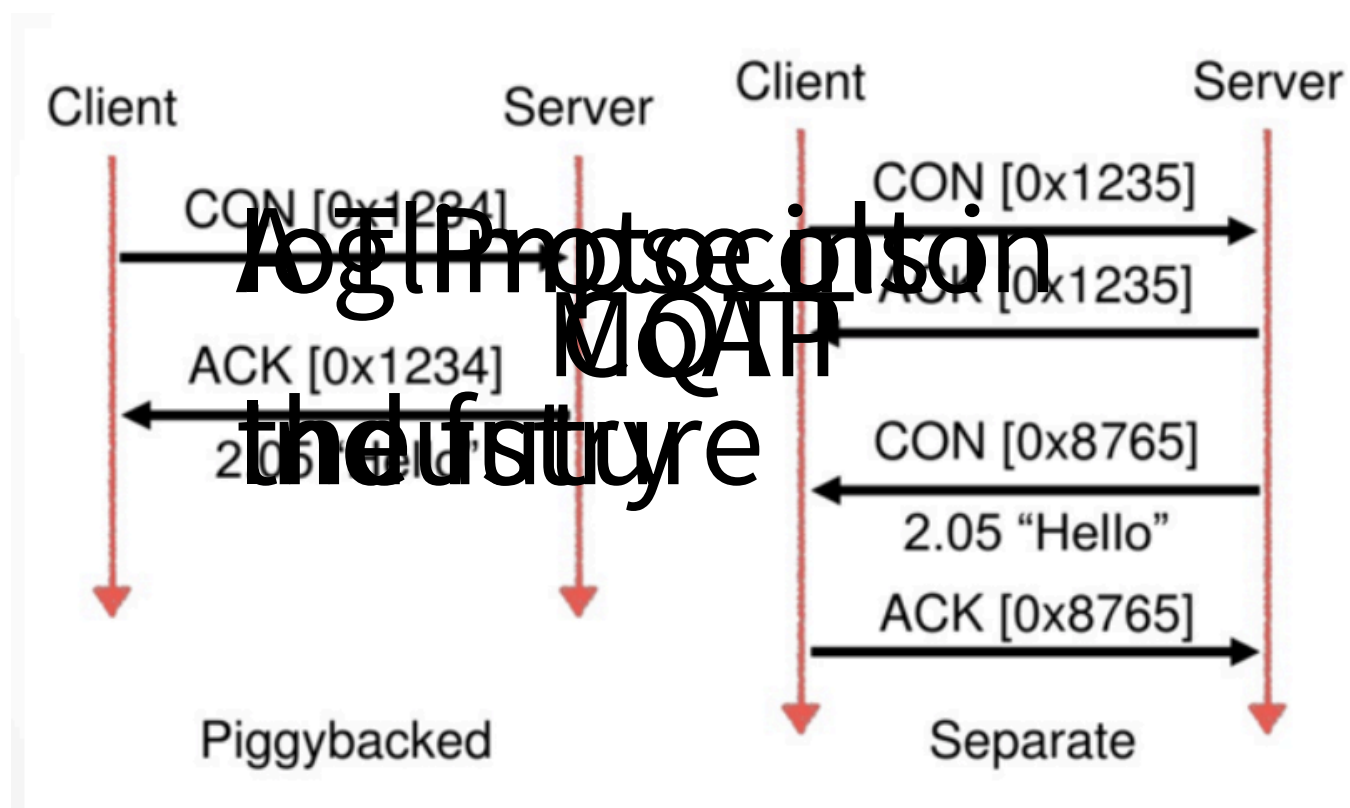
TKL - Token Length, if any, the number of Token bytes after this header

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

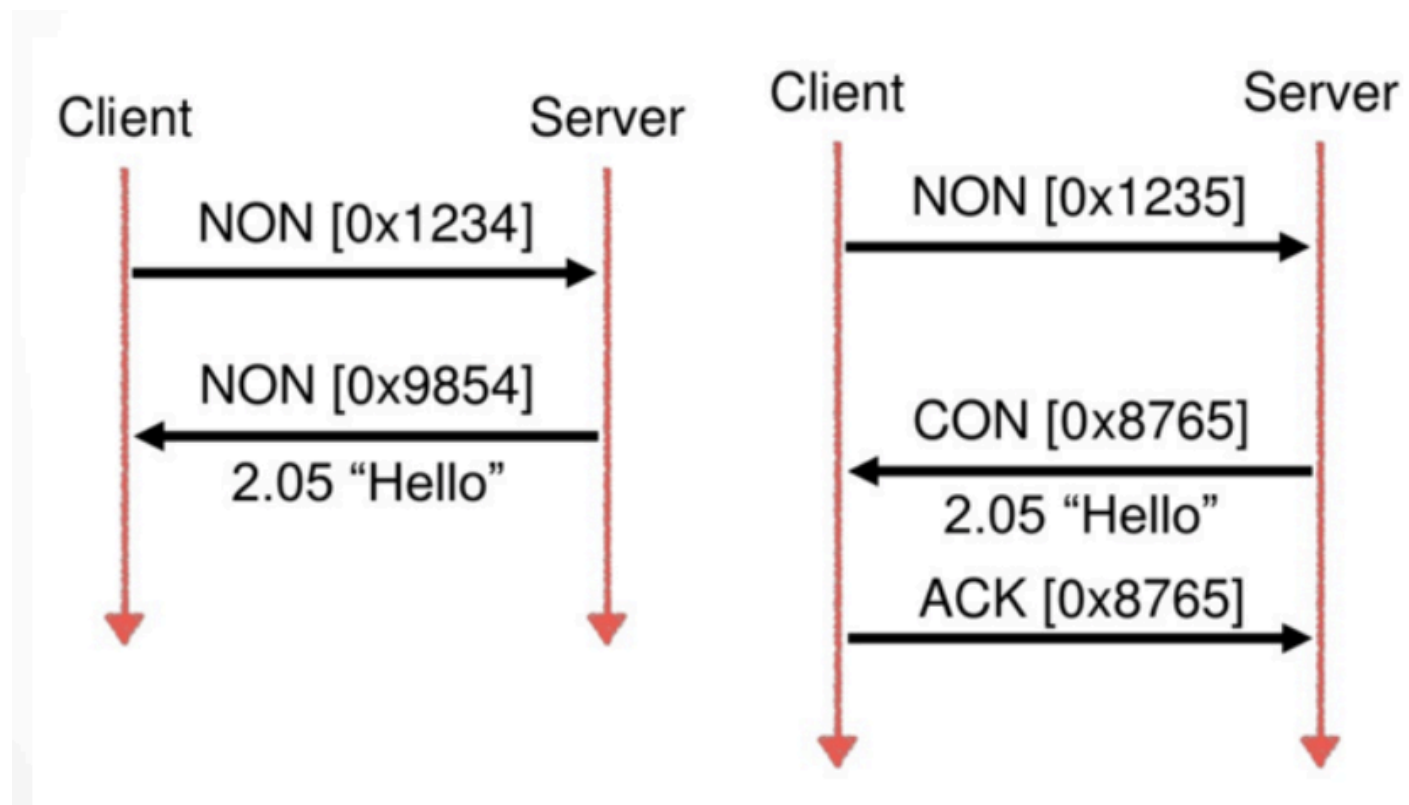
Message ID - 16-bit identifier for matching responses

Token - Optional response matching token

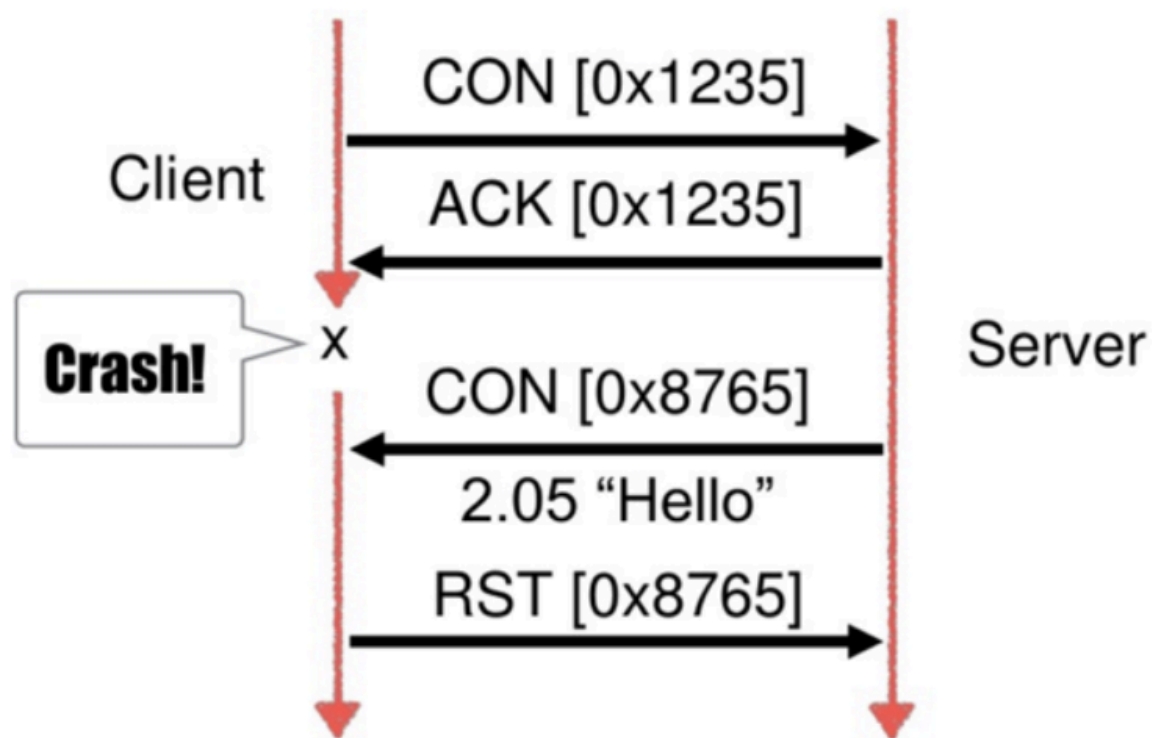
CoAP: Confirmable



CoAP: Non-confirmable



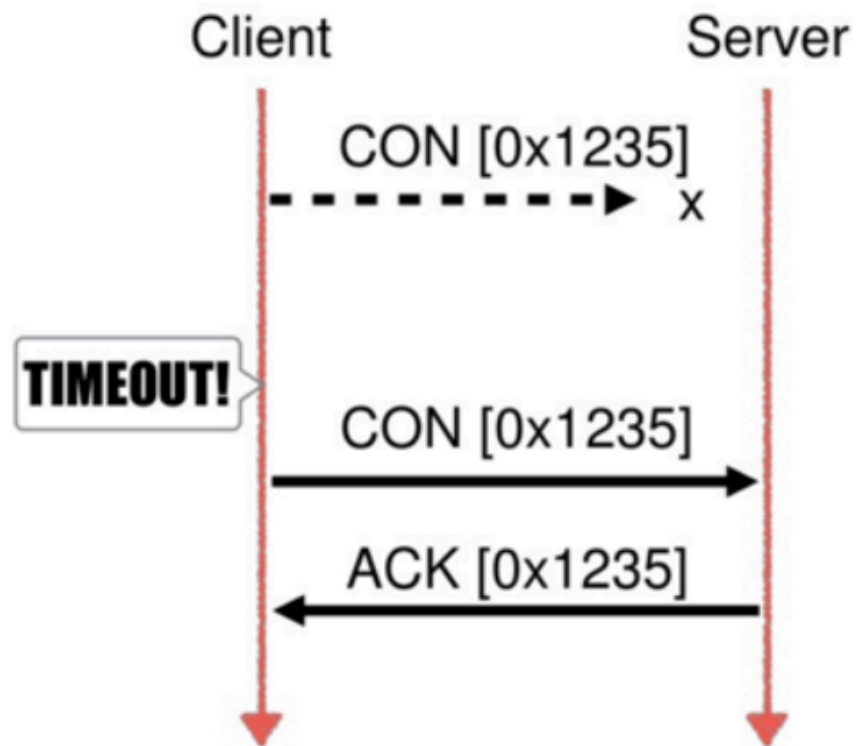
CoAP: Reset



CoAP: Reliability

- Message reliability is handled at the application layer (UDP)
- Congestion control (retransmits increase exponentially up to 247 s) — further improvements coming
- These features can be disabled, if speed is the goal

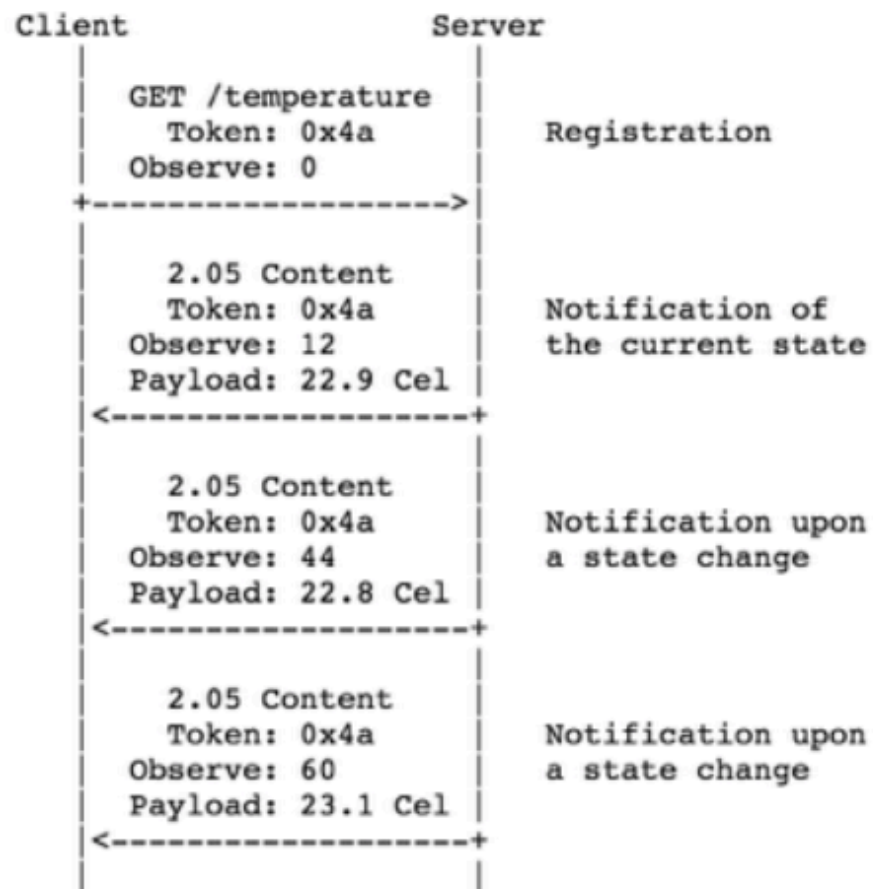
CoAP: Reliability



CoAP: Observing resources I

- Protocol extension for CoAP: RFC 7641
- Client interested in a resource over period of time
- Observer pattern
- Server ~ Client (constrained device acts as a server)

CoAP: Observing resources II



CoAP: Observing resources III

- Extension added later to the CoAP spec
- It transfers larger resource representations than can be usually accommodated in constrained networks
- Response is split in blocks
- Both sides have a say in the block size that actually will be used
- CoAP over TCP being drafted
- Authentication & authorization inspired by OAUTH and JWT

CoAP: Learn more

- General info: [**coap.technology**](http://coap.technology)
- Proposed standard:
[**tools.ietf.org/html/rfc7252**](http://tools.ietf.org/html/rfc7252)
- Copper (CoAP user-agent as a Firefox add-on):
[**github.com/mkovatsc/Copper**](https://github.com/mkovatsc/Copper)
- Several tutorials and cool features to discover

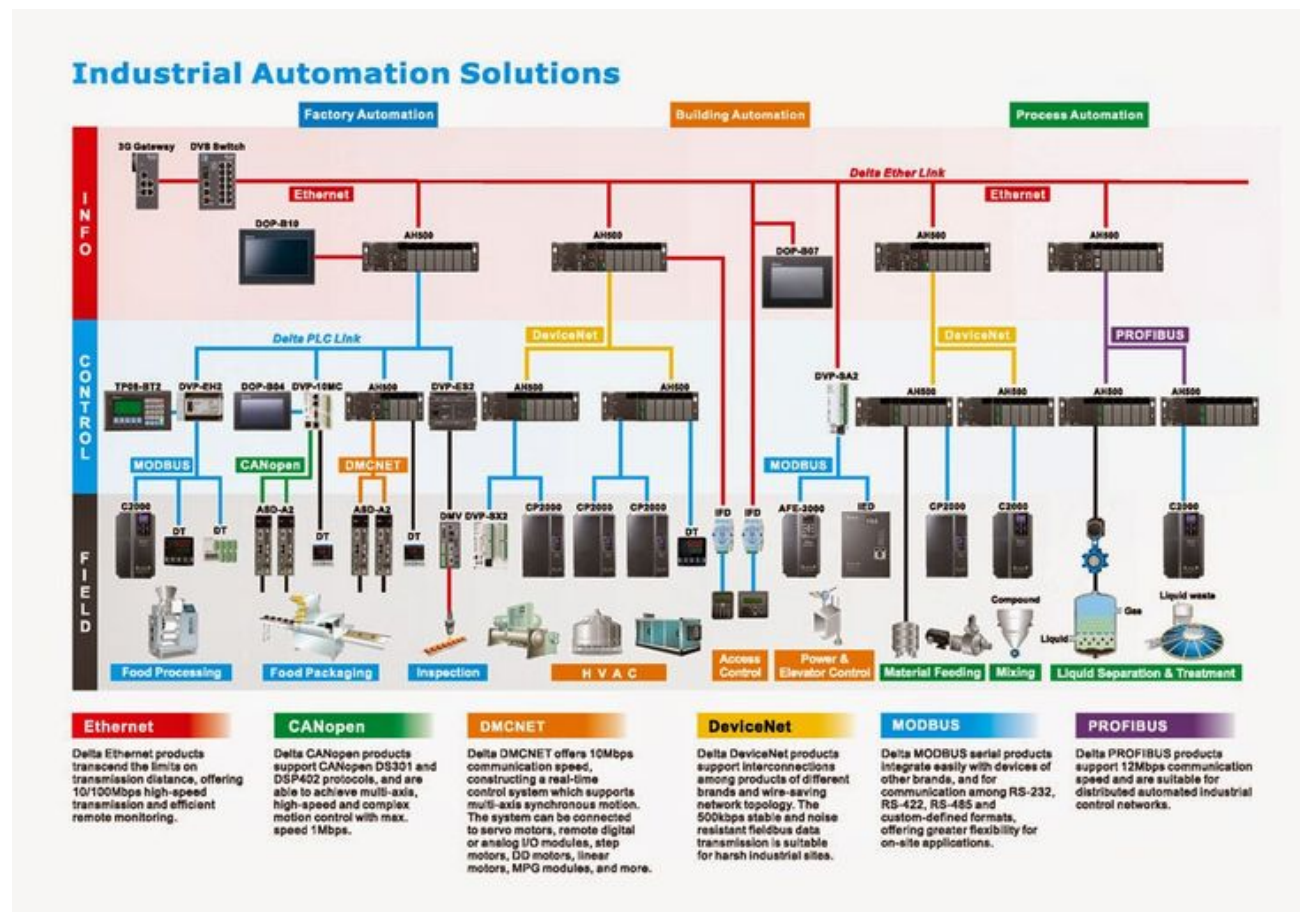


What's a Fieldbus?

"Industrial, digital bus used for real-time distributed control."

- Mainly used in **manufacturing** (assembly lines, process control, etc.)
- Connects instruments in the shop-floor
- Allows different topologies (daisy-chain, tree, etc.)
- Controllers where originally connected through serial (*just picture the cable salads*); a fieldbus solves that issue
- Cost-effective (less cables, easier maintenance)

Fieldbuses, fieldbuses all the way...



Fieldbus and Industrial Ethernet standards

- Modbus RTU (serial) (published in **1979!**)
- Modbus TCP (Ethernet)
- Profibus (serial)
- Profinet
- CAN (a *vehicle bus*, in reality)
- Bitbus
- EtherCAT
- DeviceNET
- BACnet (*Although I wouldn't call it a fielbus...*)
- And many, many more...

Just a normal PLC



What's a PFC? (with an F!) I

>_ Programmable Fieldbus Controller

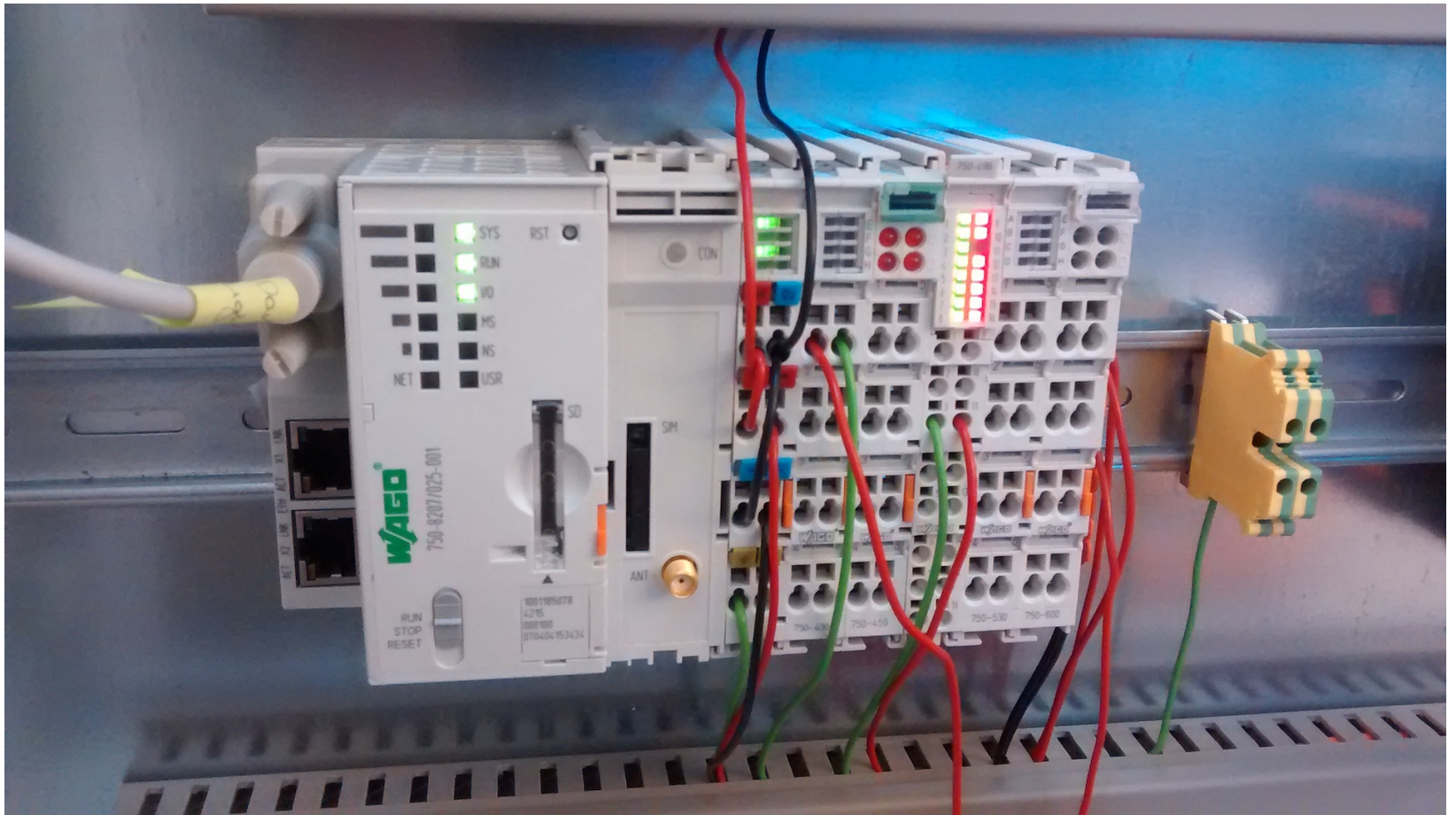
- Not to be confused with a PLC (Programmable **Logic** Controller)
- They often support different fieldbuses by default
- Easily expandable (new I/O modules can be attached, motor drivers, etc.)
- Most of them **run an OS** (and it's possible to create custom images)

What's a PFC? (with an F!) II

>_ Programmable Fieldbus Controller

- Manufacturers provide **SDKs in different languages**
- Connectivity (at least 2x network interfaces, WiFi, 3G/LTE failover, serial, etc.)
- More and more have **built-in HTTP APIs**
- Most of them integrate web servers for configuration, HMIs or SCADA functionalities
- They may replace the proprietary automation software suites

What's a PFC?



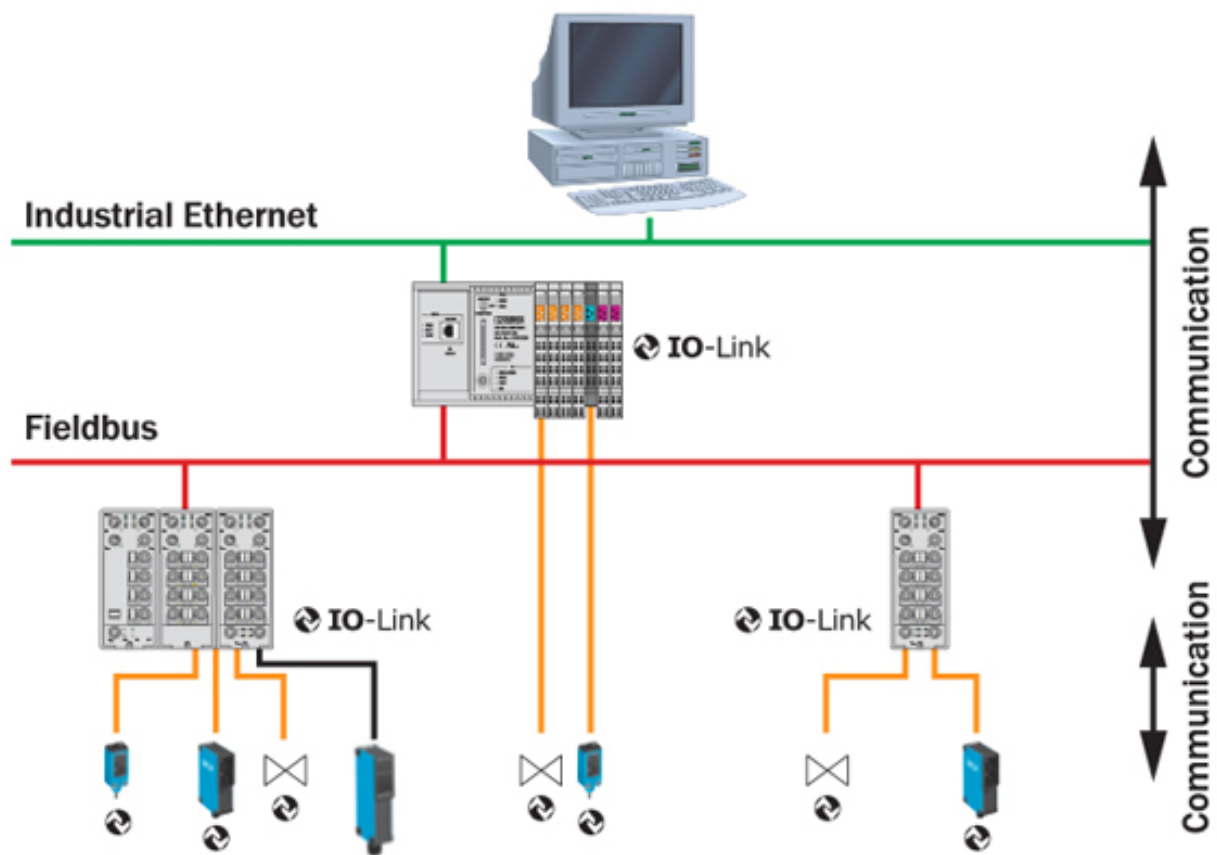
IO-Link: *State of the art*

"IO-Link is the first standardised IO technology worldwide (IEC 61131-9) for the communication with sensors and actuators.

***IO-Link is no fieldbus** but the further development of the existing, tried-and-tested connection technology."*

- 3-wire connections
- Smart sensors: they work out-of-the-box, and they "identify" themselves
- Descriptive files that include information of the manufacturer, the type, and the calibration
- Sensors can be replaced, and the calibration and specs file updated remotely
- IO-Link masters support different fieldbus and Industrial Ethernet standards

IO-Link: Topology example



IO-Link: Master / sensor aggregator

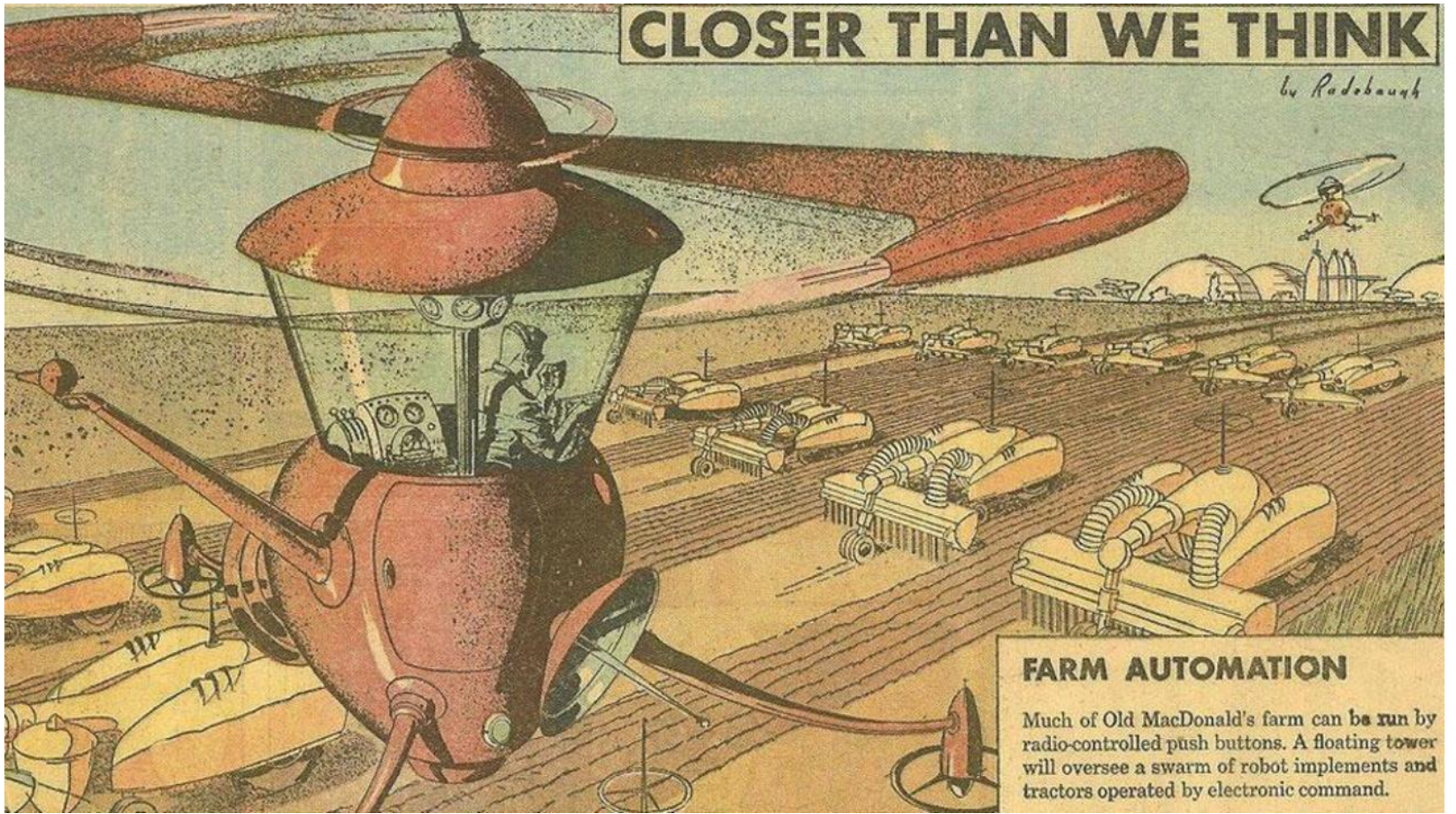


And now ... subtle product placement



CLOSER THAN WE THINK

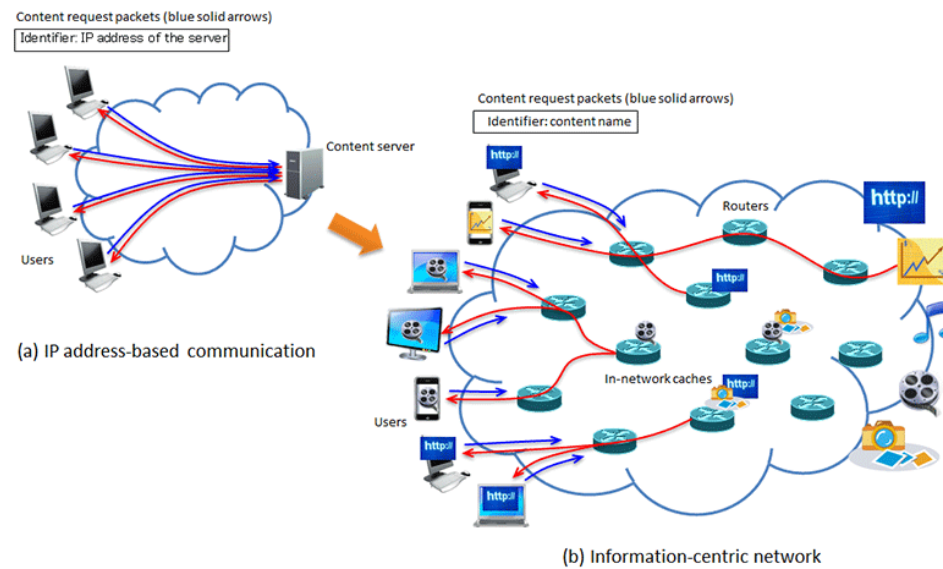
by Rodabaugh



FARM AUTOMATION

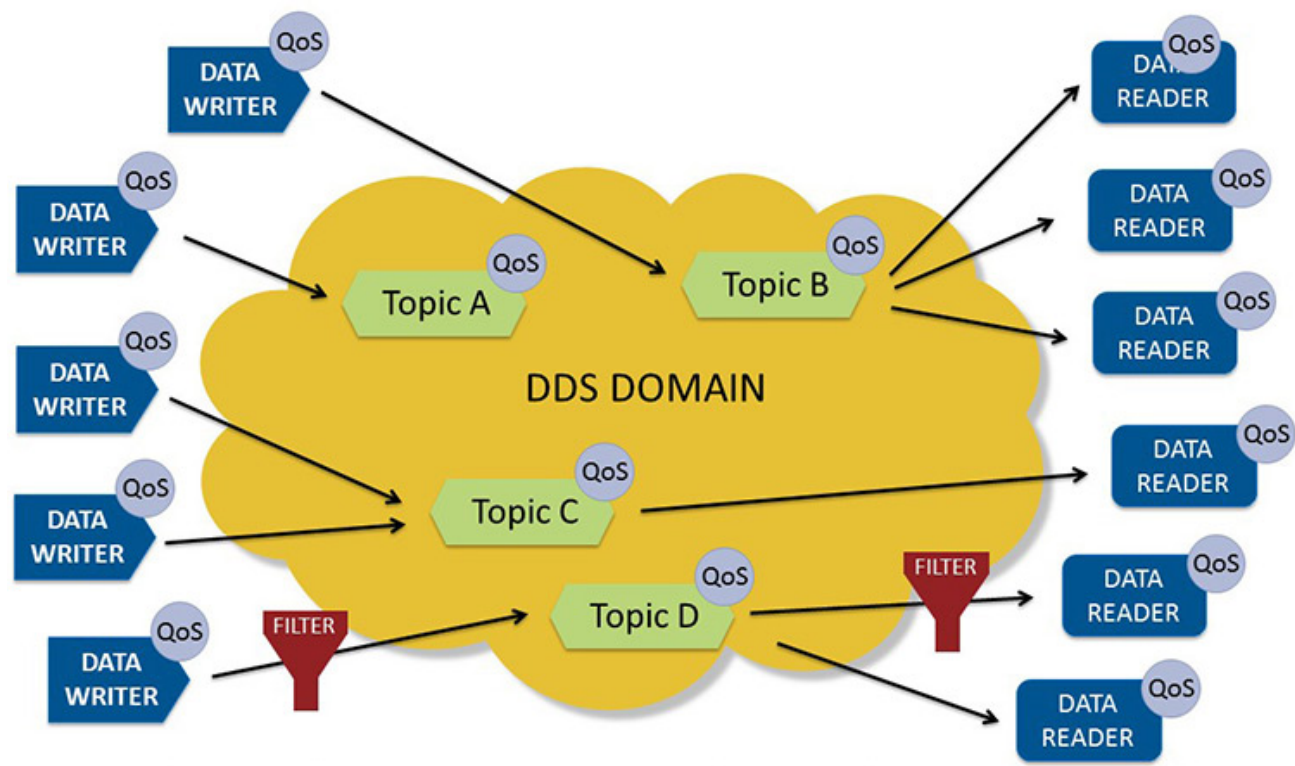
Much of Old MacDonald's farm can be run by radio-controlled push buttons. A floating tower will oversee a swarm of robot implements and tractors operated by electronic command.

Information Centric Networking (ICN) vs Host Centric Networking (HCN)



- HCN: Conversation between **hosts** — **who** to talk to.
- ICN: Spreads **data** objects — **what** to say

Data Distribution Service (DDS)



DDS in a nutshell

- Has been around for some time — DDS 1.0 (2005).
- Main entities:
 - *Domain Participant*
 - *Data Writer*
 - *Publisher*
 - *Data Reader*
 - *Subscriber*
 - *Topic*
- All networking is abstracted. Usually implemented on top of raw sockets
- Anycasting and Multicasting

ICN in a nutshell I

- Shares *packet forwarding* with IP mostly
- Outline of request - response
 - **Consumer** requests **named data**: **Interest**
 - **Interest** is forwarded to a place (or places) where named data exists
 - **Forwarder** records the interface on which the Interest was received
 - **Data** is returned in a **Content** message
 - Data in Content is **signed** to avoid tampering
- A lot of caching strategies possible — see the web

ICN in a nutshell II

- **Communication** between **consumers** and **named data**
- **Forwarders** interact with messages and maintain a state per-message (!== IP)
- **Data name** instead of IP address
- Anycasting and Multicasting
- **Consumer** can roam — easy mobility

ICN in a nutshell III

- Is a research topic
- Many open questions:
- Routing
 - *Congestion control*
 - *Push (event) also, not only polling*
- Multiple research projects: US, Europe
- Watch this space

Conclusions

- There are many type of networks
- The protocol to use **depends** on what part of **which** network you are
- **Opinionated** Cheatsheet:
 - *MQTT: D2C or C2C*
 - *CoAP: D2D*
 - *HTTP/1.1: C2C*
 - *HTTP/2: C2C — possibly D2C*
 - *DDS: C2C & D2C — claims of D2D seem exaggerated*
 - *WebSockets: C2C*
- Operational Technology (OT) will be around for a long time
- No silver bullet

About us

>_ Antonio

- [GitHub](#)
- [LinkedIn](#)

>_ Jaime

- [GitHub](#)
- [LinkedIn](#)

>_ This presentation

- tinyurl.com/gotoams2017-iot

Questions?

