A gentle introduction to IoT protocols with security in mind

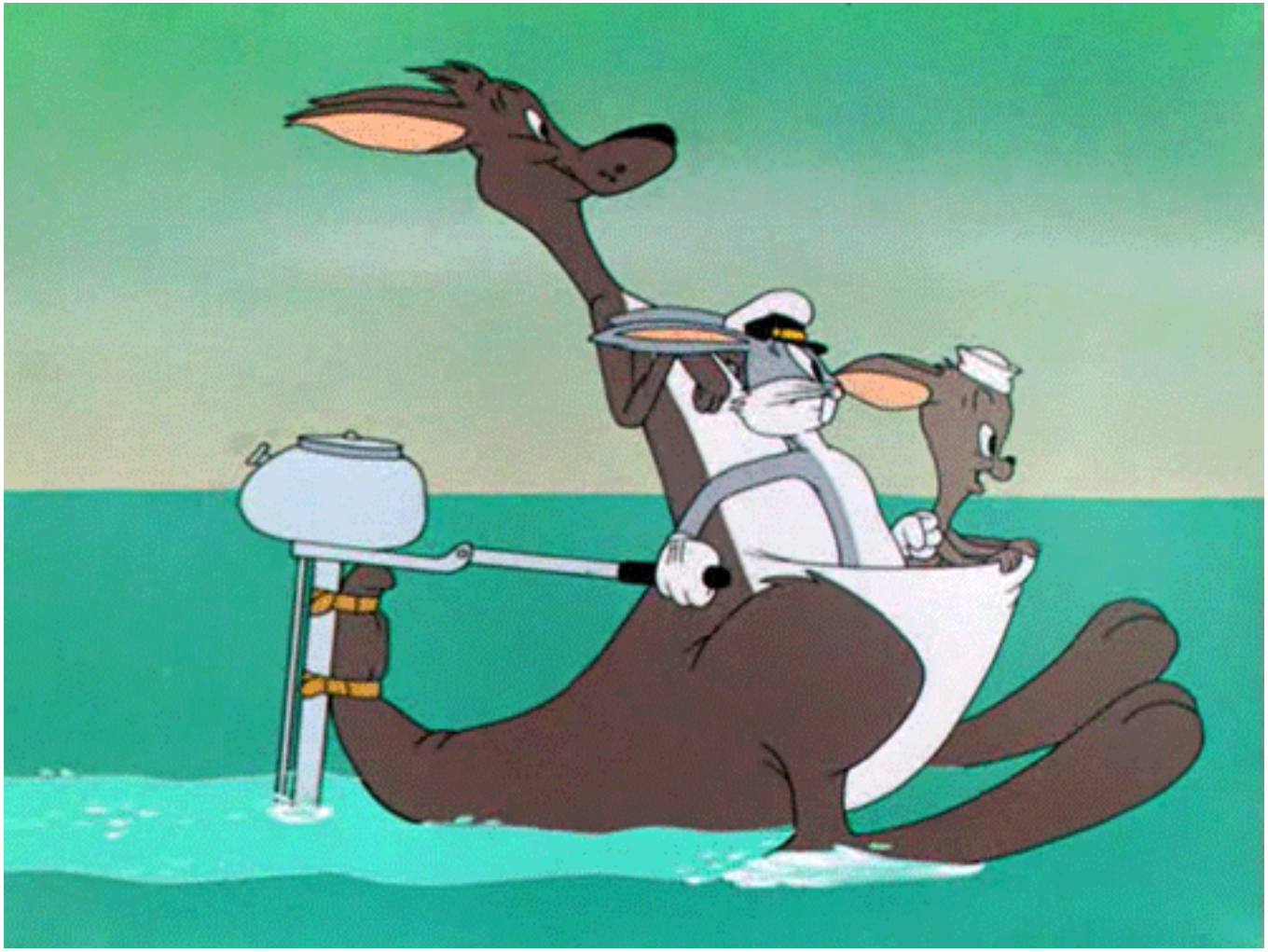
Genetec Connect'19, Montreal

António Almeida

December 2018

relayr.

warning: we' ll go fast!



relayr.

On-line Man-Computer Communication, 1962

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.

The Computer as a Communication Device - J.C.R. Licklider &

Robert Taylor, 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.

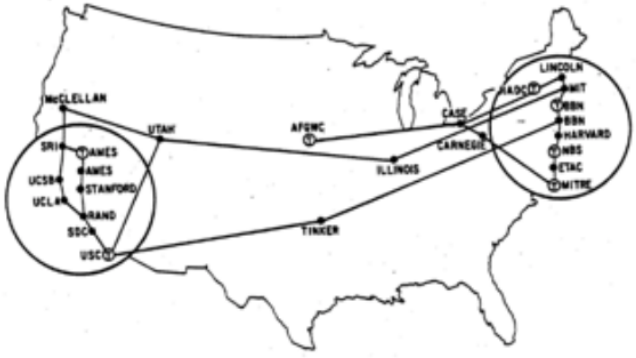
That is a rather startling thing to say, but it is our conclusion. As if in confirmation of it, we participated a few weeks ago in a technical meeting held through a computer. In two days, the group accomplished with the aid of a computer what normally might have taken a week.

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

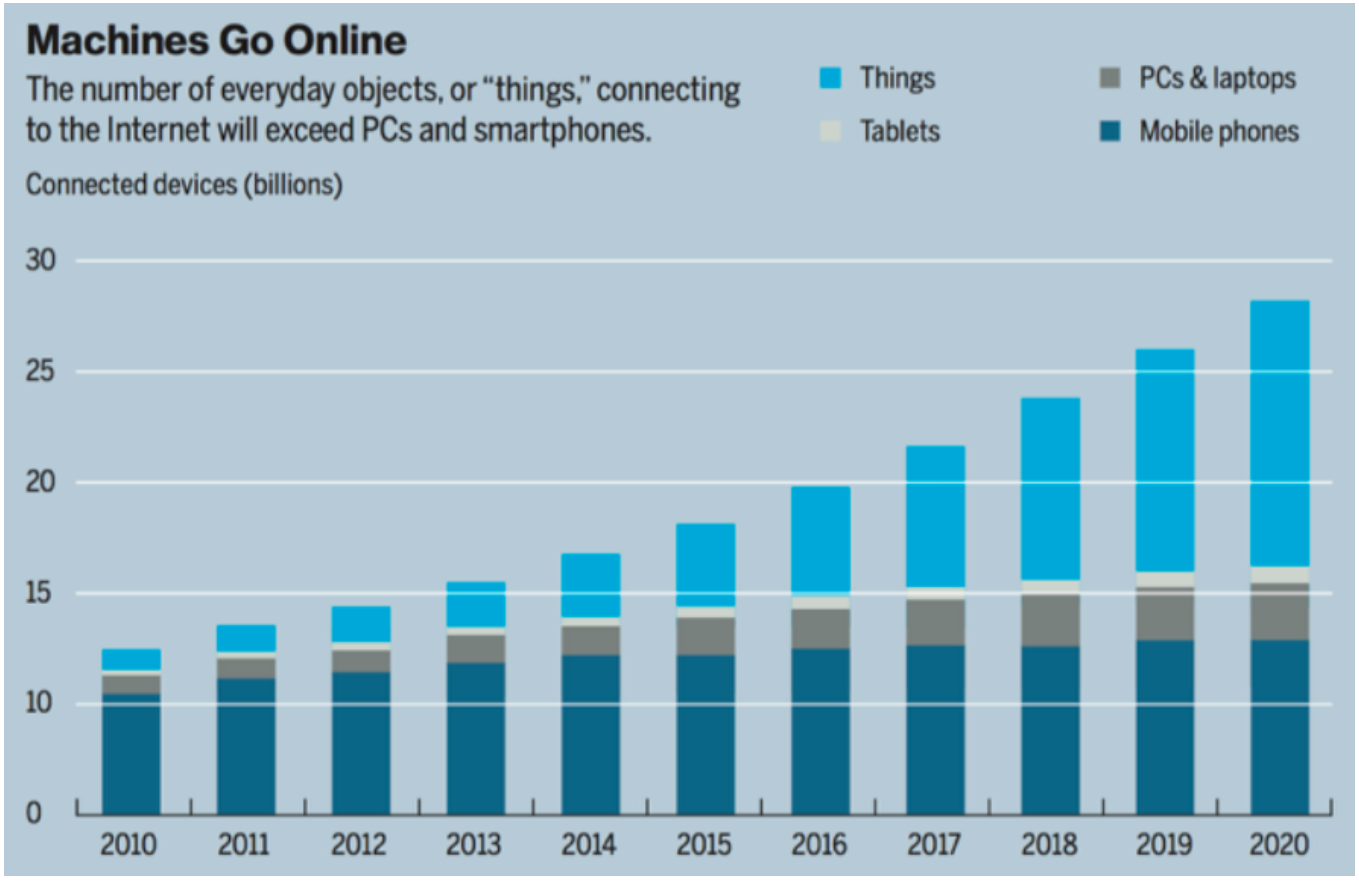
Juli 1977

relayr.

why more protocols?

relayr.

connected devices I



MIT Technology Review, 2014

relayr.

connected devices II

- 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

relayr.

device cost & connectivity

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

relayr.

constrained devices

- IETF Definition: `tools.ietf.org/html/rfc7228`
- limited processing power
- unreliable networking
- low power *(so they can run on batteries)*

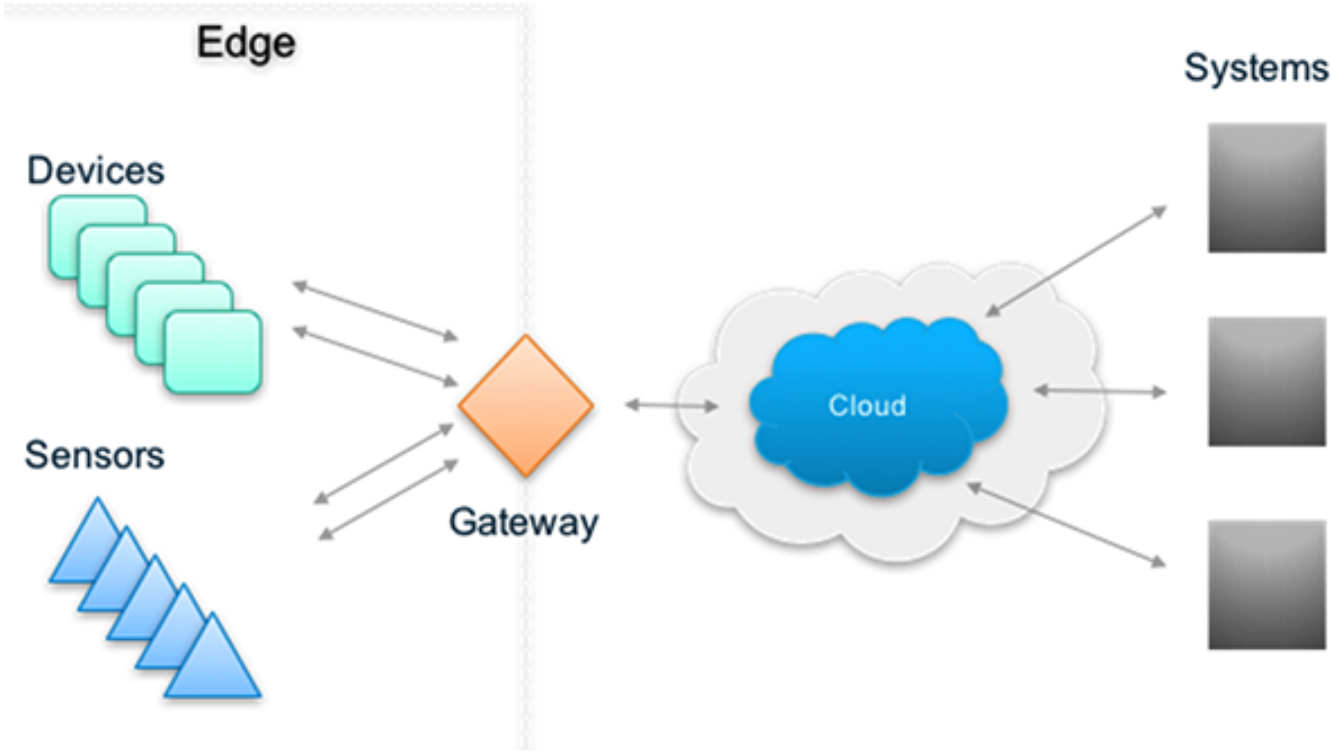
relayr.

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.”

relayr.

edge has devices – cloud has servers



relayr.

edge

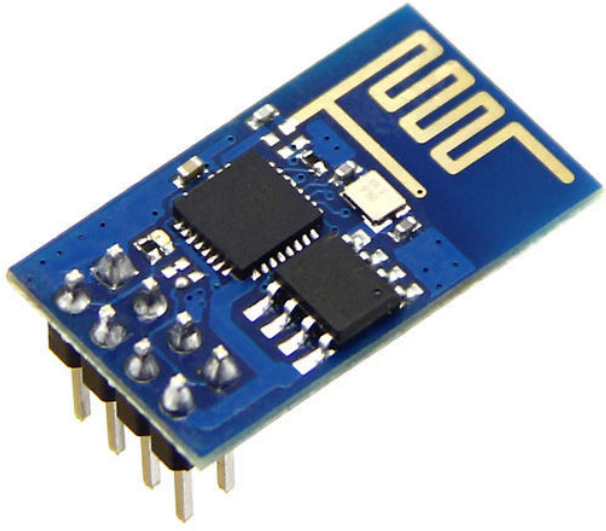
devices $\neq$ gateways

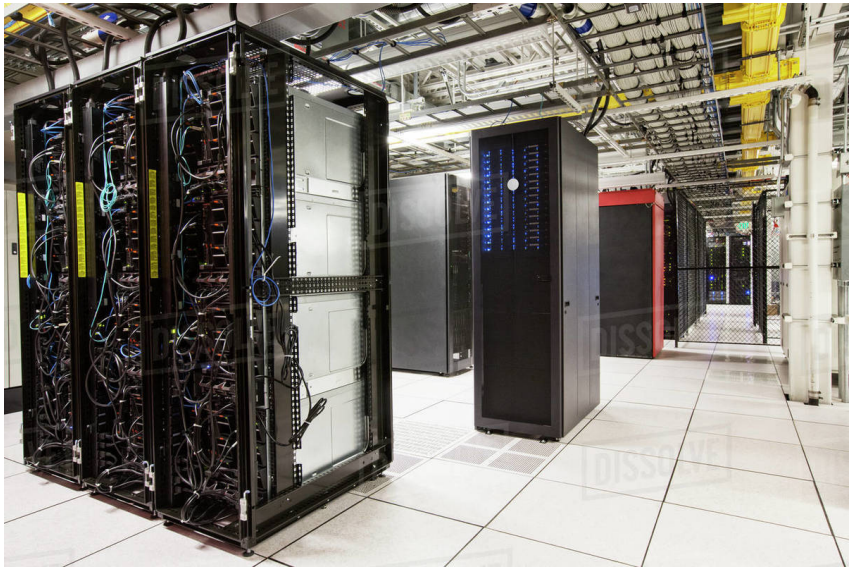
- devices talk to other devices northbound and southbound
- gateways talk to the cloud northbound and devices southbound
- device to device (d2d)
- device to cloud (d2c)

security considerations I: logo soup

relayr.

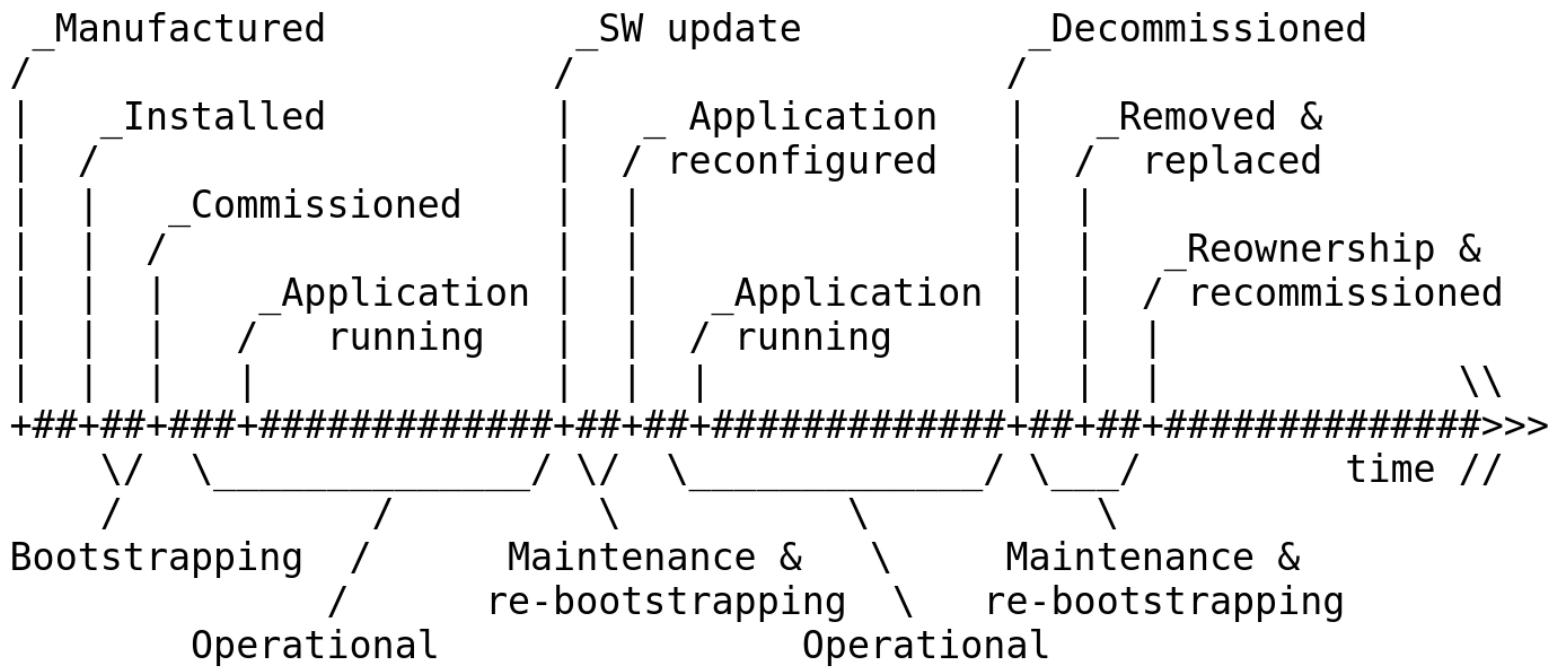
security considerations II: system heterogeneity





relayr.

security considerations III: Thing lifecycle



relayr.

further reading on IoT security

State-of-the-Art and Challenges for the Internet of Things Security

https://datatracker.ietf.org/doc/draft-irtf-t2trg-iot-seccons/?include_text=1

- provides in depth discussion of IoT security considerations
- references many other IETF working groups
- inspiration for IoT security comes mostly from web
- takes into account the fact that the devices are much more constrained
- critical applications

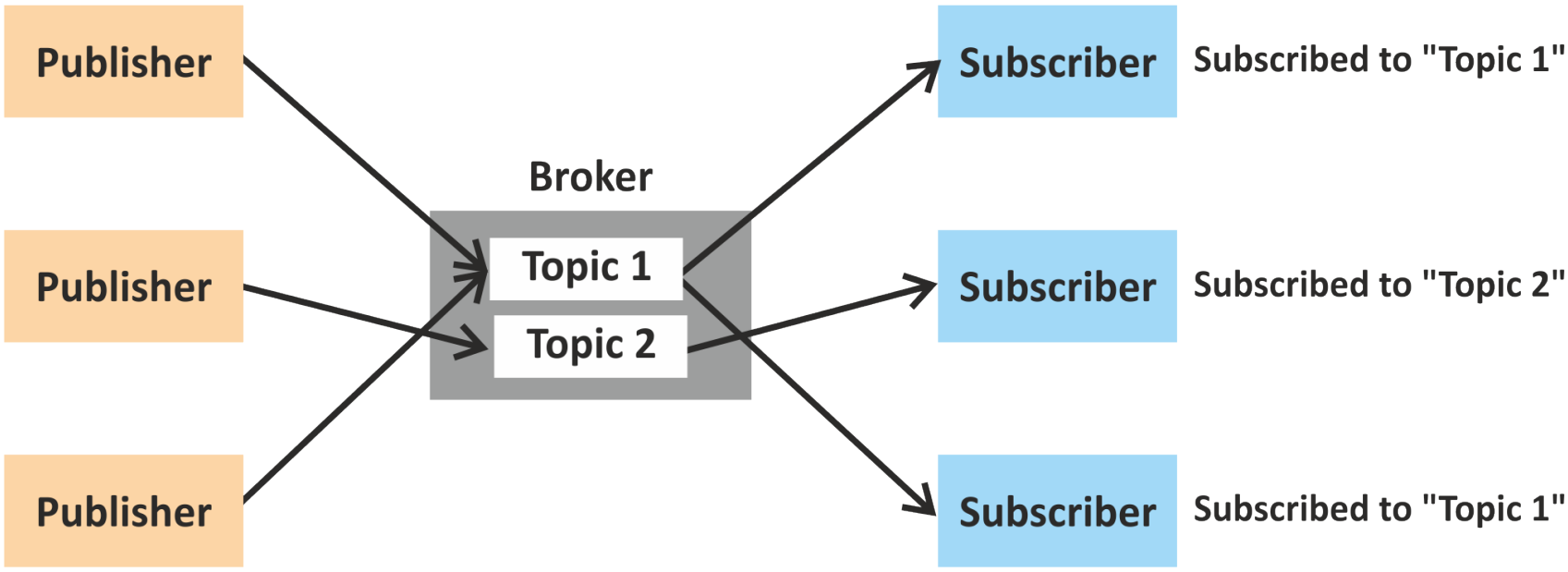
MQTT I: basics

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 II: publish-subscribe model



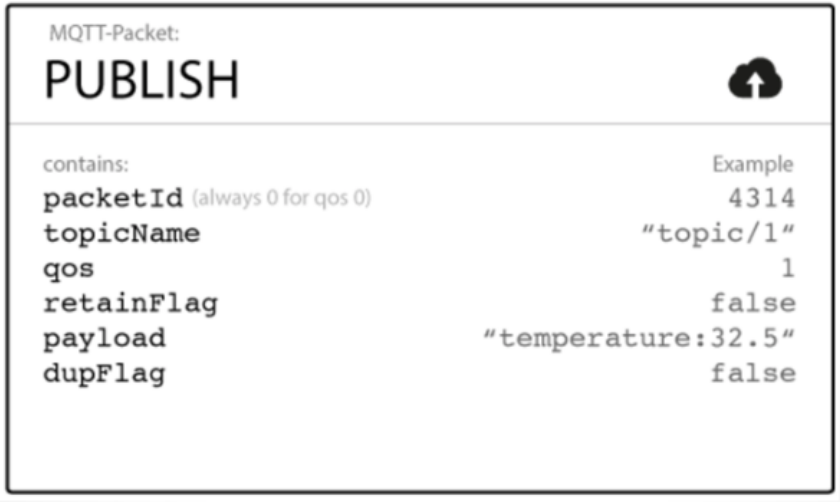
relayr.

MQTT III: 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

relayr.

MQTT IV: publishing to a topic



relayr.

MQTT V: 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)

relayr.

MQTT VI: 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 VII: 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

relayr.

MQTT security highlights

- MQTTS: MQTT over TLS
- Basic Authentication like in HTTP (password/user)
- broker constitutes Single Point of Failure (SPOF)

relayr.

MQTT 5: what's new highlights I

- jump from 3.1.1 to 5 is due to single byte for version in frame
- metadata can now be embedded in a published message: user properties
- reason codes: kind-of-status codes for CoAP and HTTP
- shared subscriptions: client load balancing

MQTT 5: what's new highlights II

- optional `Content-Type` header (MIME)
- message format: 1 means UTF8 encoded, 0 other type of data
- topic aliasing
- `AUTH` packet for other authentication schemes besides basic
- request/response communication pattern

MQTT 5: state of implementations I

- VerneMQ: currently stable version supports most of MQTT5 features
- written in Erlang: high performance
- supports server side Lua scripting
- *probably* the most standards compliant implementation out there
- free software: available on github

MQTT 5: state of implementations II

- the C written mosquitto (mosquitto.org) has now a branch where some of the version 5 features are being implemented
- lagging VerneMQ
- requires re-architecting the mosquitto library that is used in many sofware projects

relayr.

MQTT VIII: 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
- lots of good tutorials out there

CoAP I: basics

A specialized web transfer protocol for use with constrained nodes and constrained networks.

RFC 7252 - Constrained Application Protocol

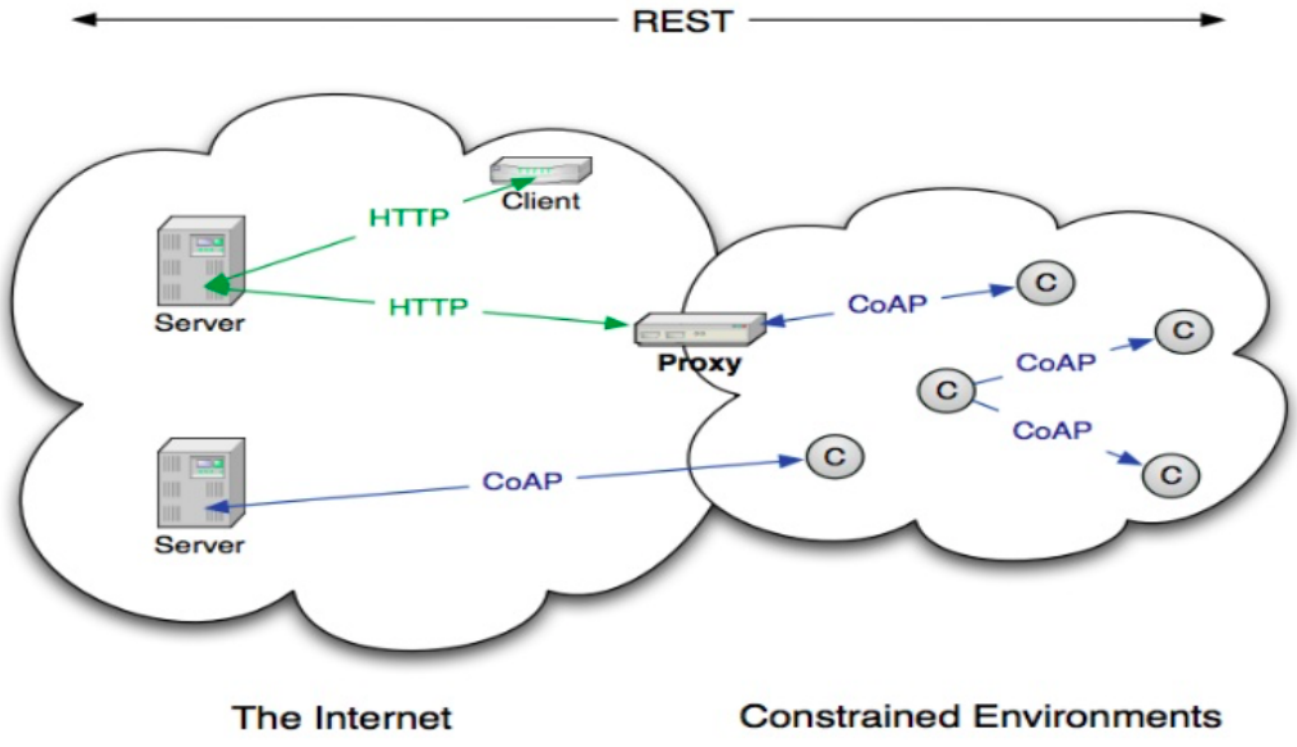
relayr.

CoAP development

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

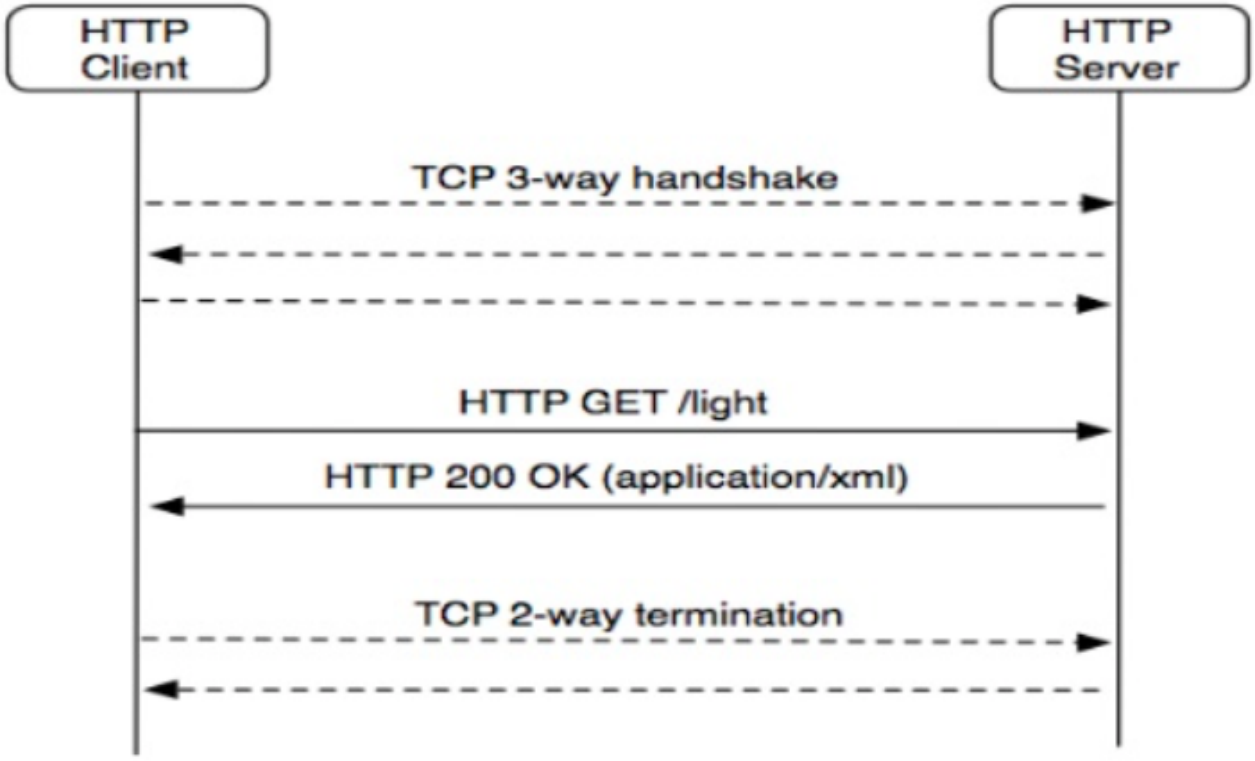
relayr.

CoAP II: RESTful environment



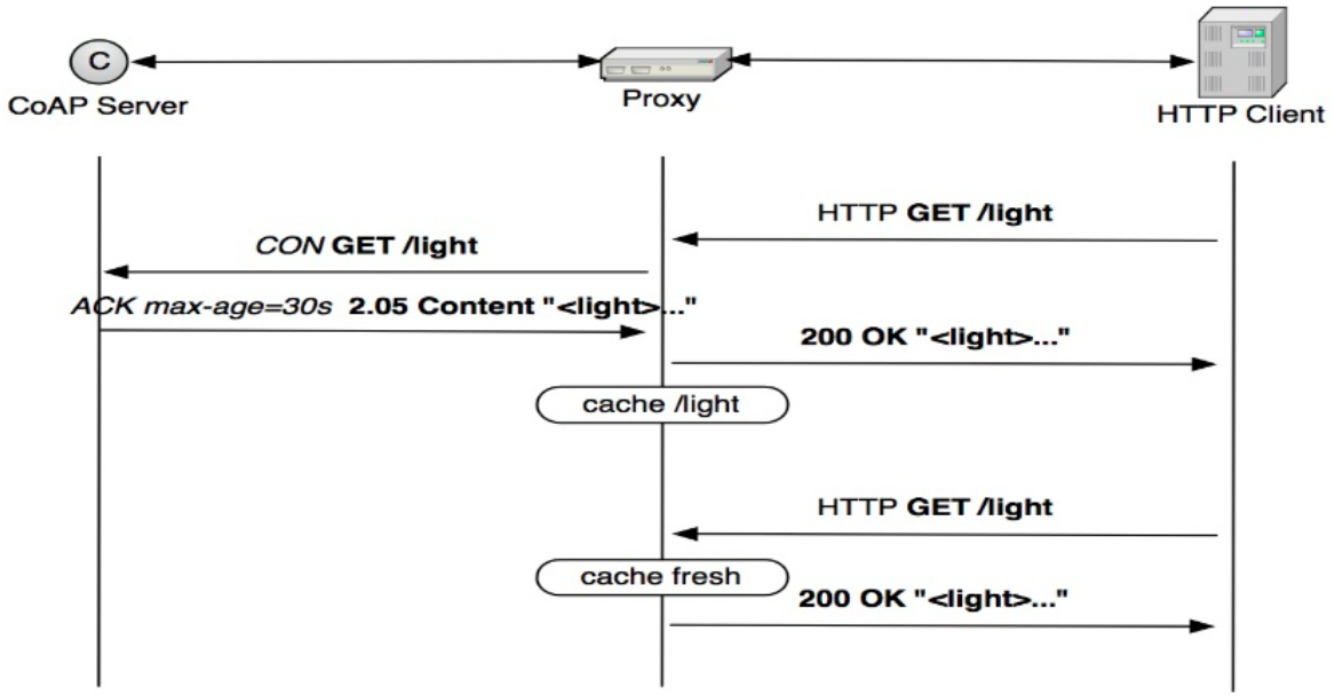
relayr.

CoAP III: inspired by HTTP



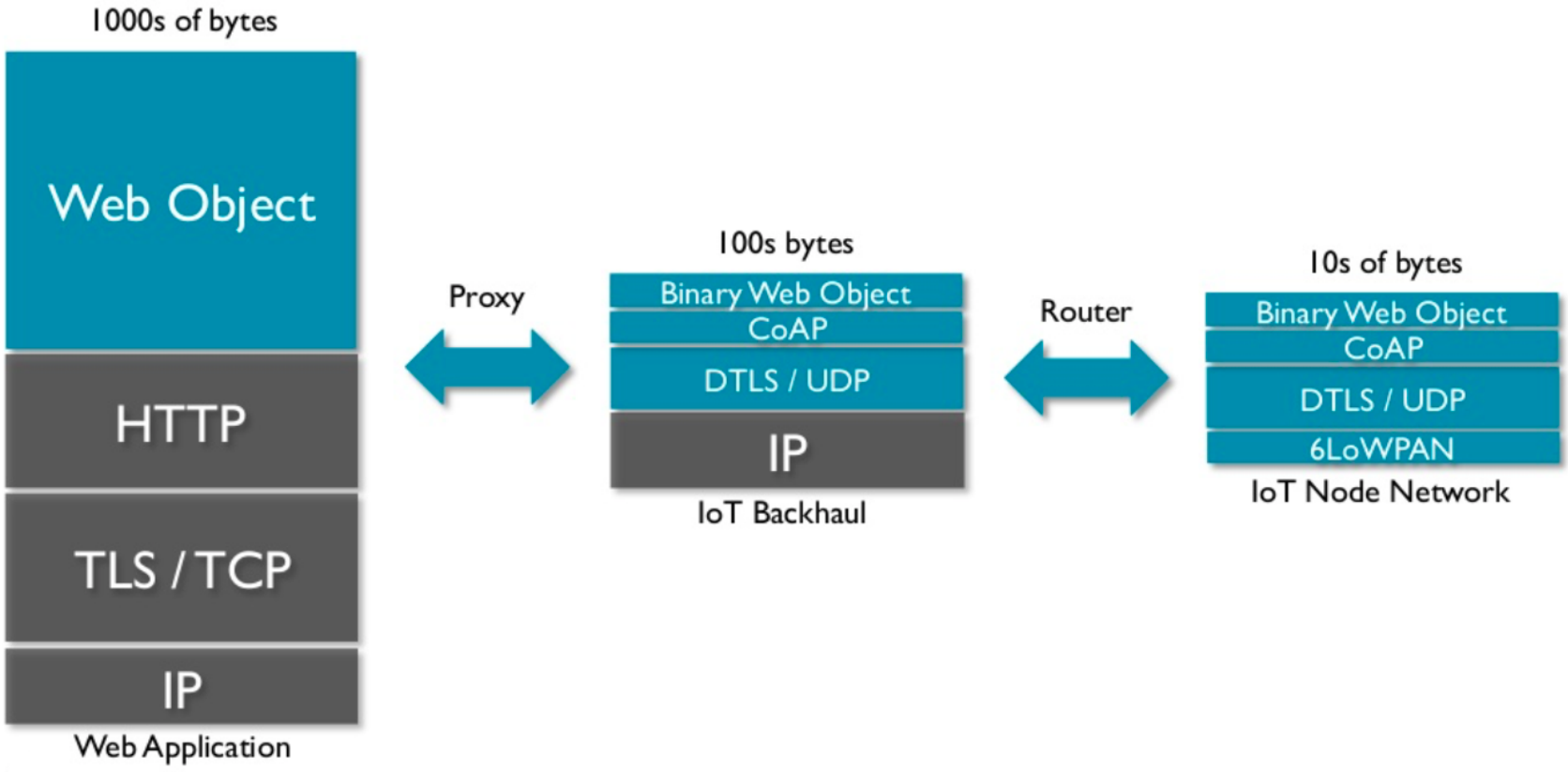
relayr.

CoAP IV: proxying with HTTP



relayr.

CoAP V: HTTP vs CoAP



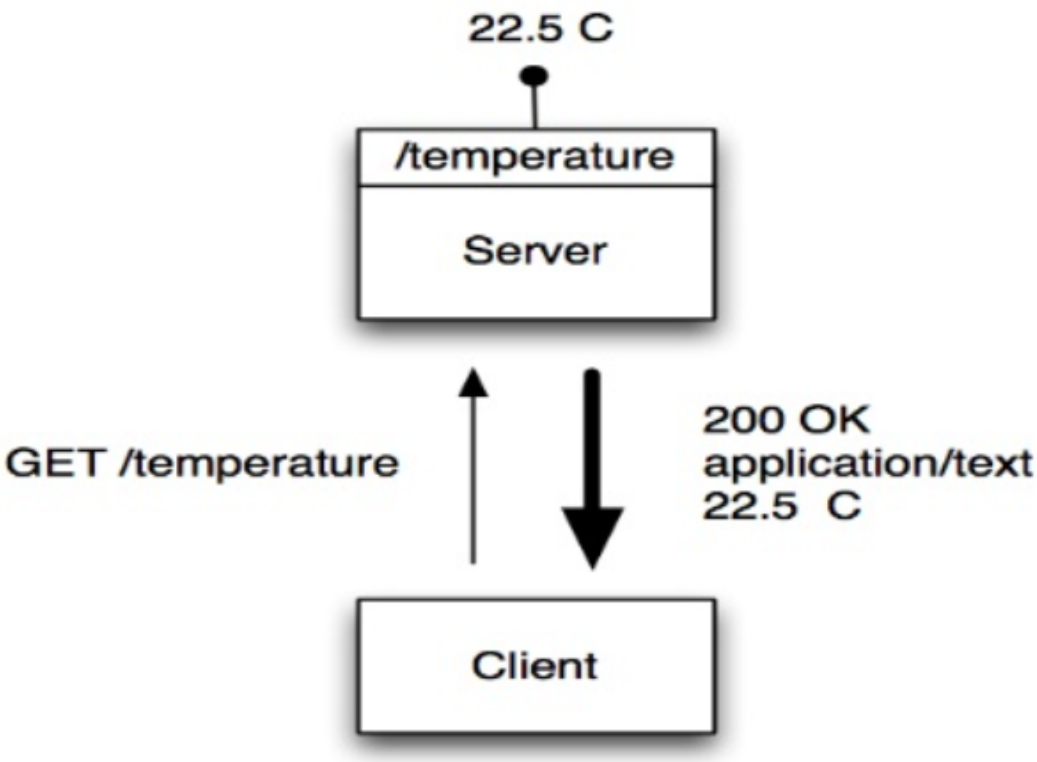
relayr.

CoAP VI: Functionalities

- URI
- `GET / POST / PUT / DELETE / PATCH / FETCH / iPATCH`
- Content-Type support (XML, JSON, CBOR,…)
- built-in discovery — `.well-known/core`
- multicast support* asynchronous message exchanges
- designed to be extensible
- congenial to IPv6

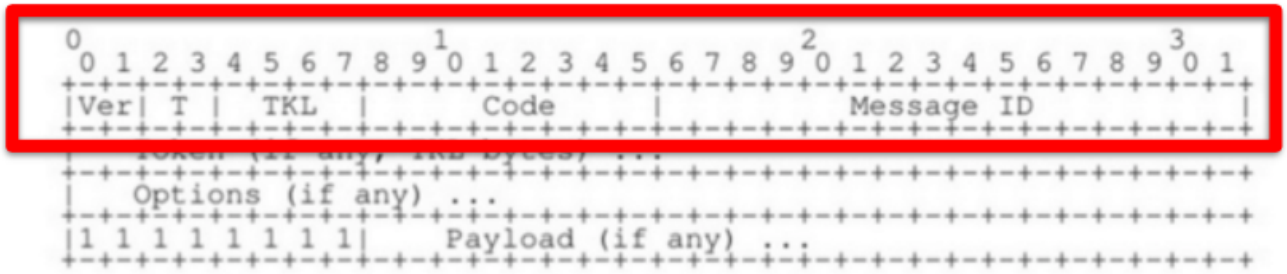
relayr.

CoAP VII: client / server



relayr.

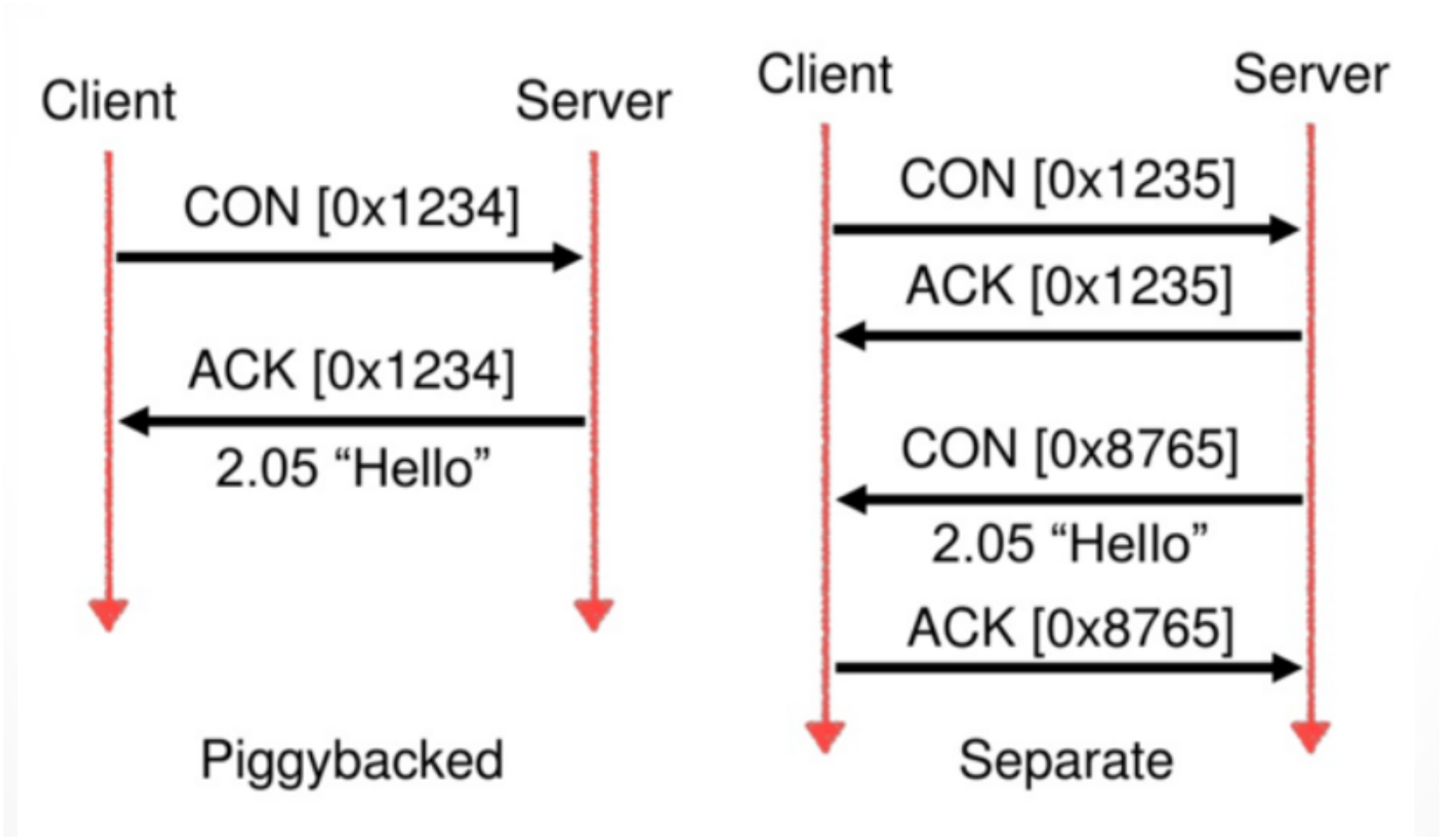
CoAP VIII: message format



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

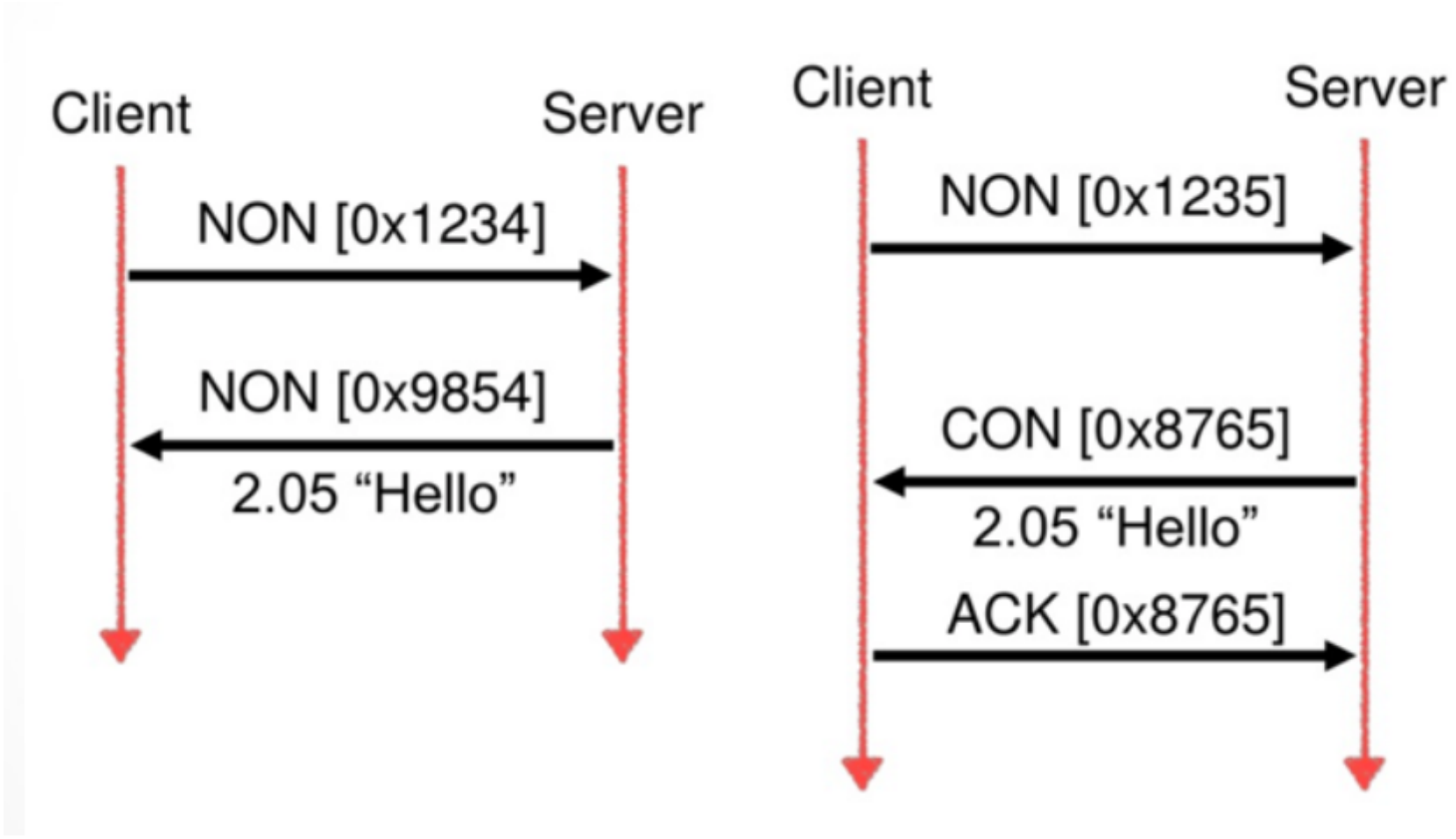
relayr.

CoAP IX: confirmable messages



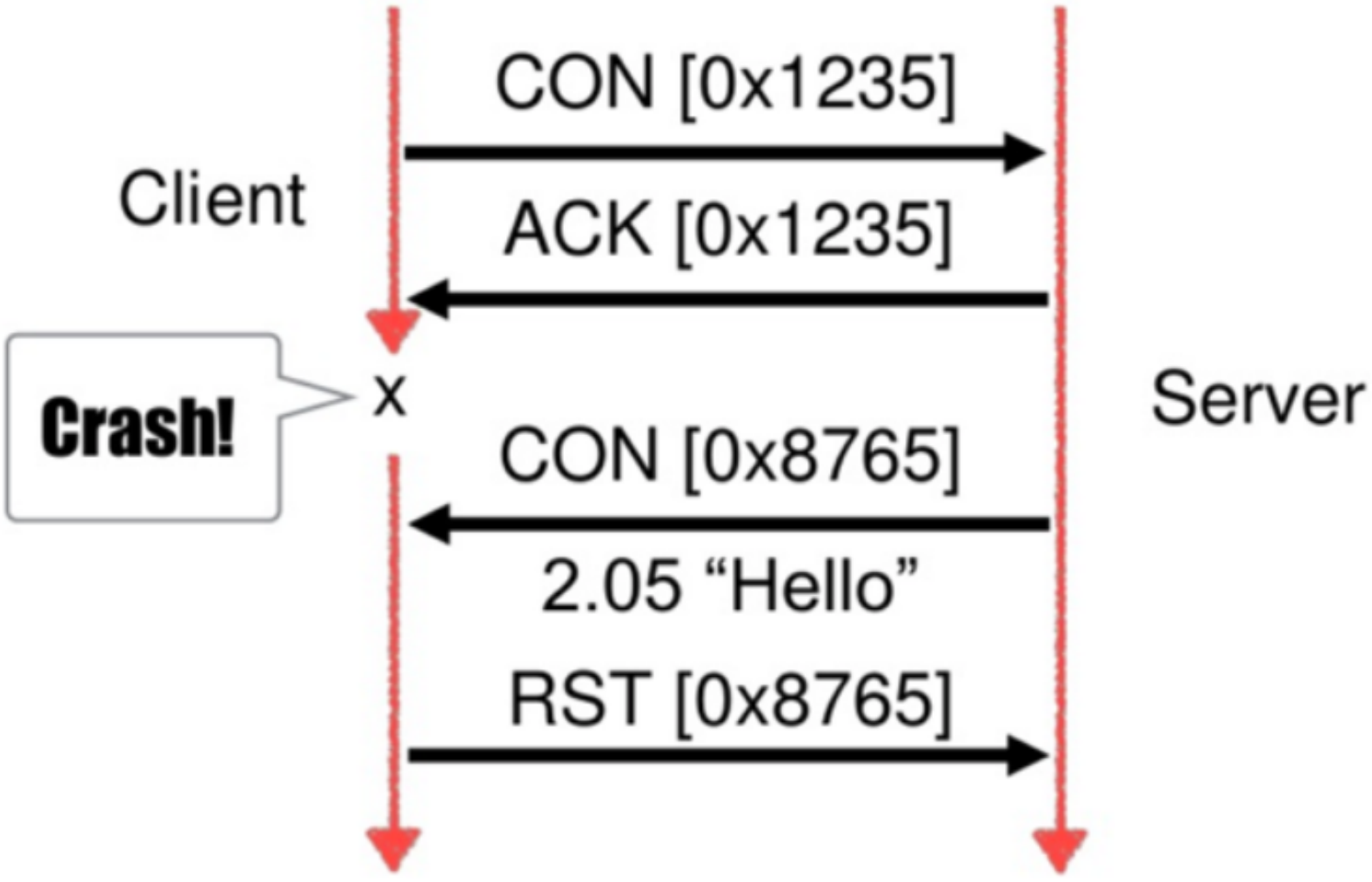
relayr.

CoAP X: non-confirmable messages



relayr.

CoAP XI: reset

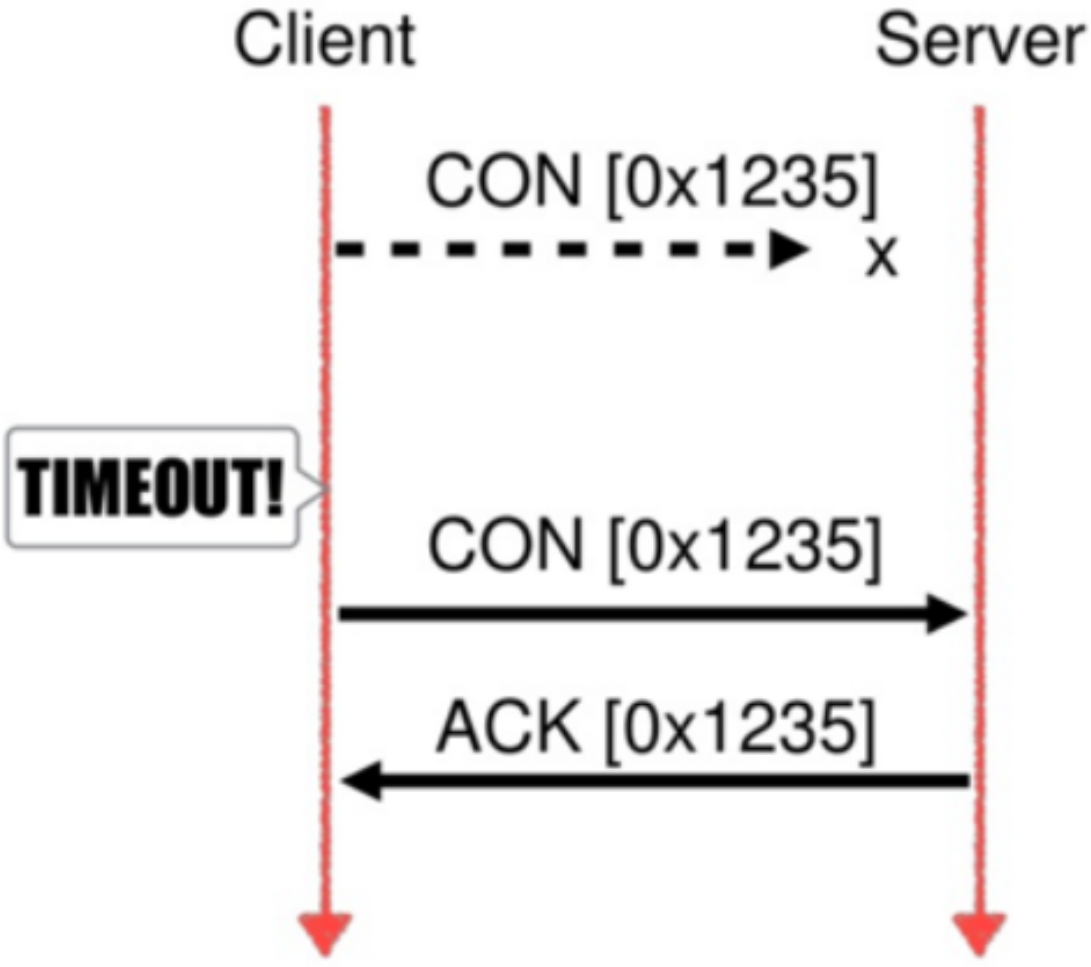


relayr.

CoAP XII: 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 XIII: reliability continued

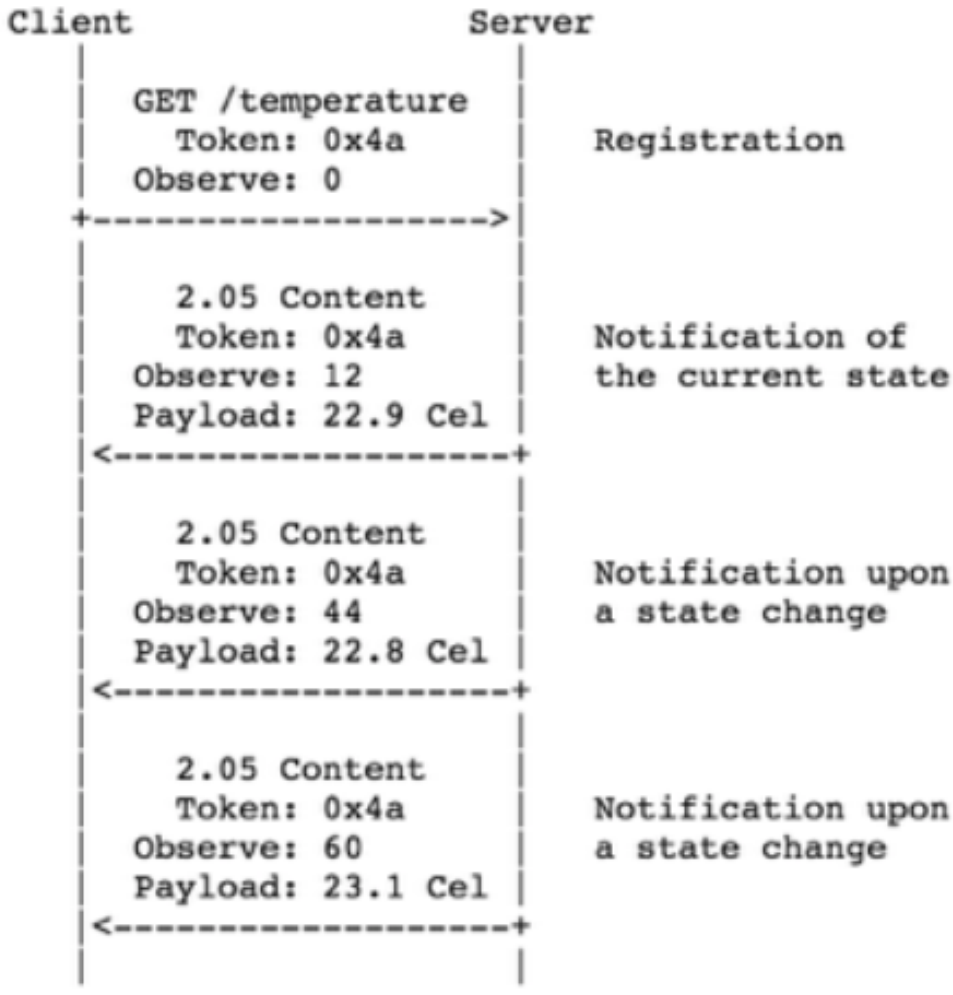


relayr.

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



relayr.

CoAP observing resources III

- extension added later to the CoAP spec
- 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 used

relayr.

CoAP security highlights I

- CoAP over DTLS
- multiple ways to distribute secret keys: pre-shared, asymmetric unverified and certificate based
- ongoing work to make certificate based more amenable to constrained devices
- recent CoAP over TCP RFC

CoAP security highlights II

- Authorization relies in not so constrained devices being delegated by more constrained ones
- Authentication and Authorization for Constrained Environments IETF working group
- authentication & authorization inspired by OAUTH and JWT

relayr.

CoAP: improving on HTTP ideas - FETCH & iPATCH

- standard introduced in RFC 8132 — April 2017
- FETCH makes it possible to have complex GET like operations w/o relying in long query strings or a *fake* POST
- iPATCH is an idempotent way to PATCH a resource
- few implementations of FETCH and iPATCH

relayr.

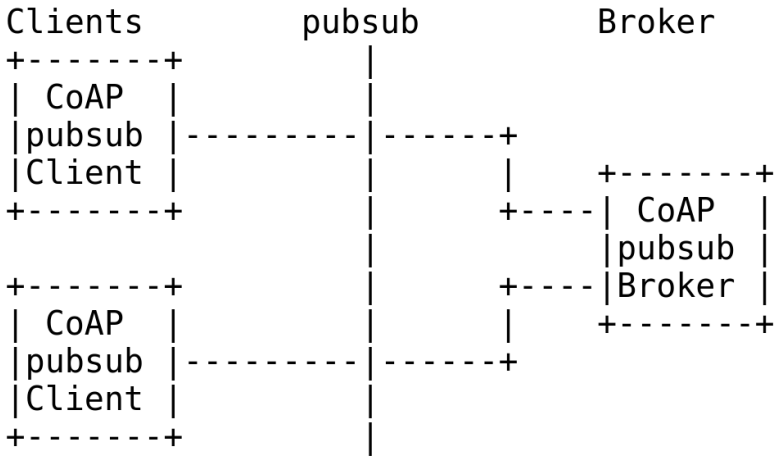
CoAP pubsub architecture I

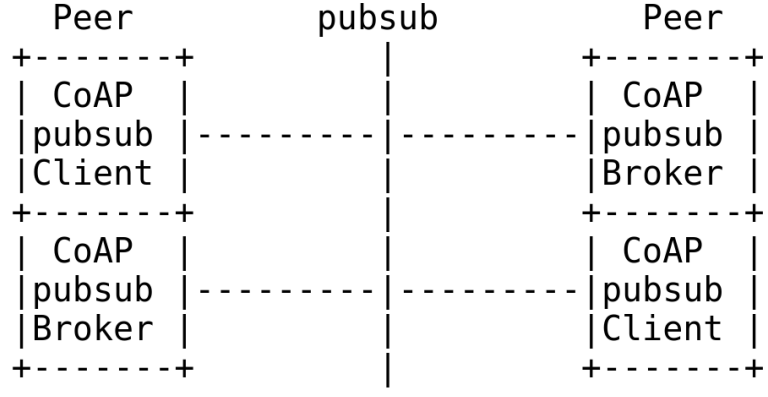
- working draft:

  https://tools.ietf.org/html/draft-koster-core-coap-pubsub-05
- CoAP brokers forward data from some nodes to others
- broker and brokerless based approaches

relayr.

CoAP pubsub architecture II





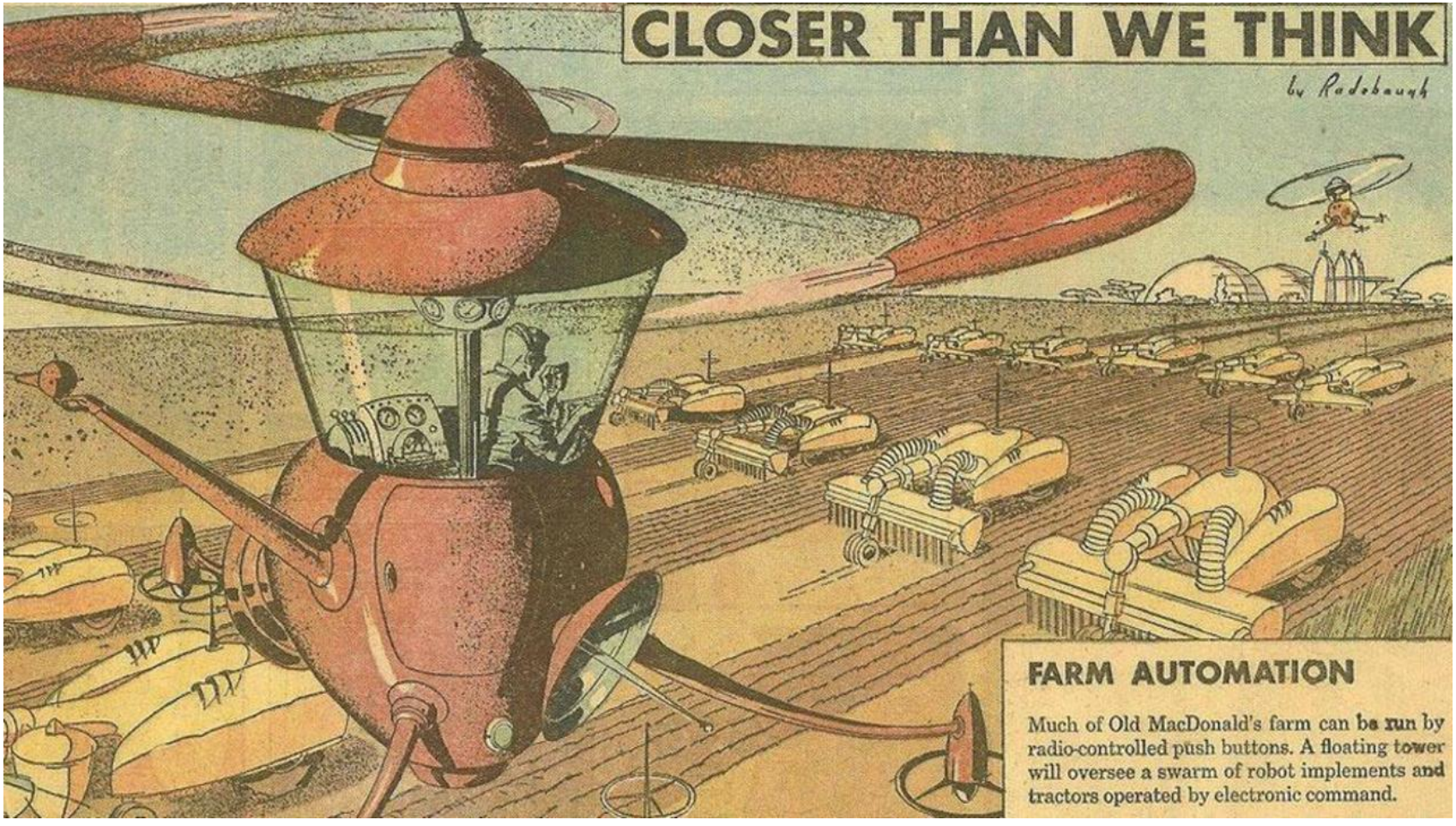
relayr.

learn more about CoAP

- general info: `coap.technology`
- proposed standard: `tools.ietf.org/html/rfc7252`
- copper (CoAP user-agent ~~as a Firefox add-on~~):

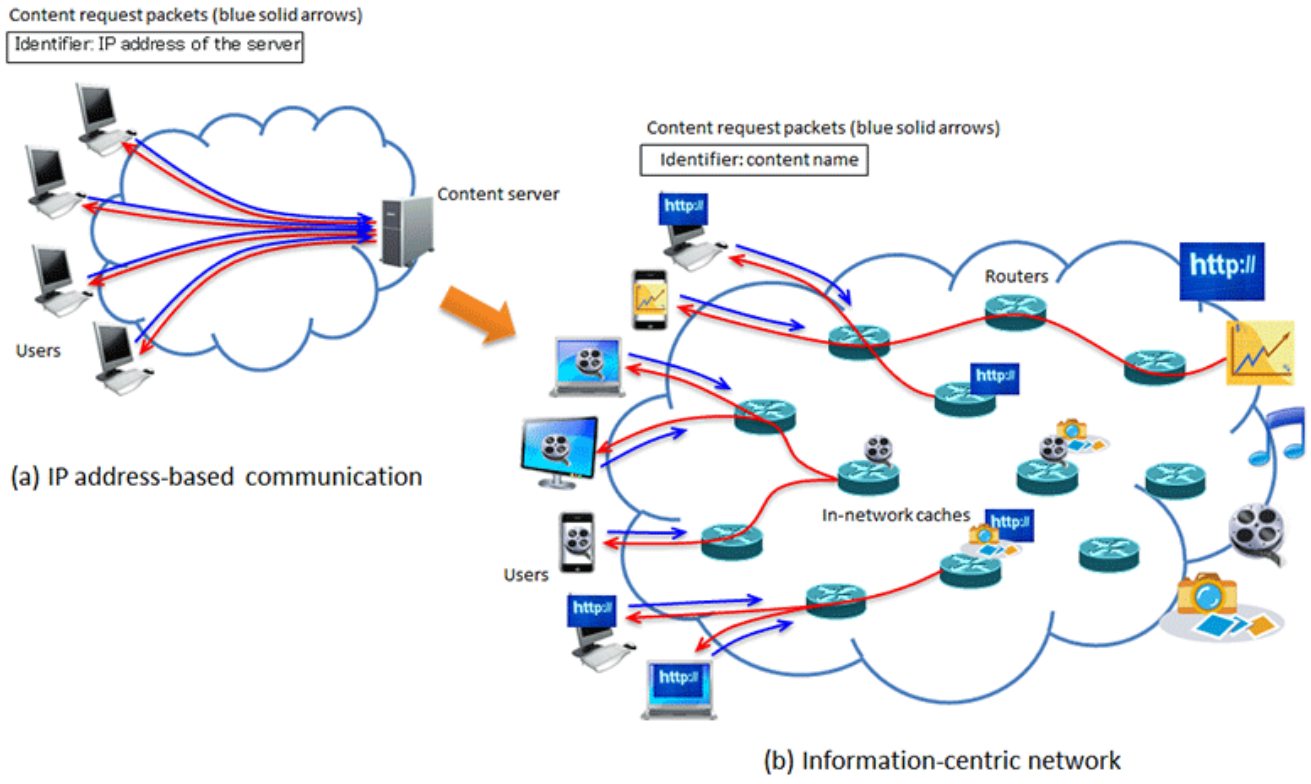
  `github.com/mkovatsc/Copper`
- several tutorials and cool features to discover

a glimpse into the future



relayr.

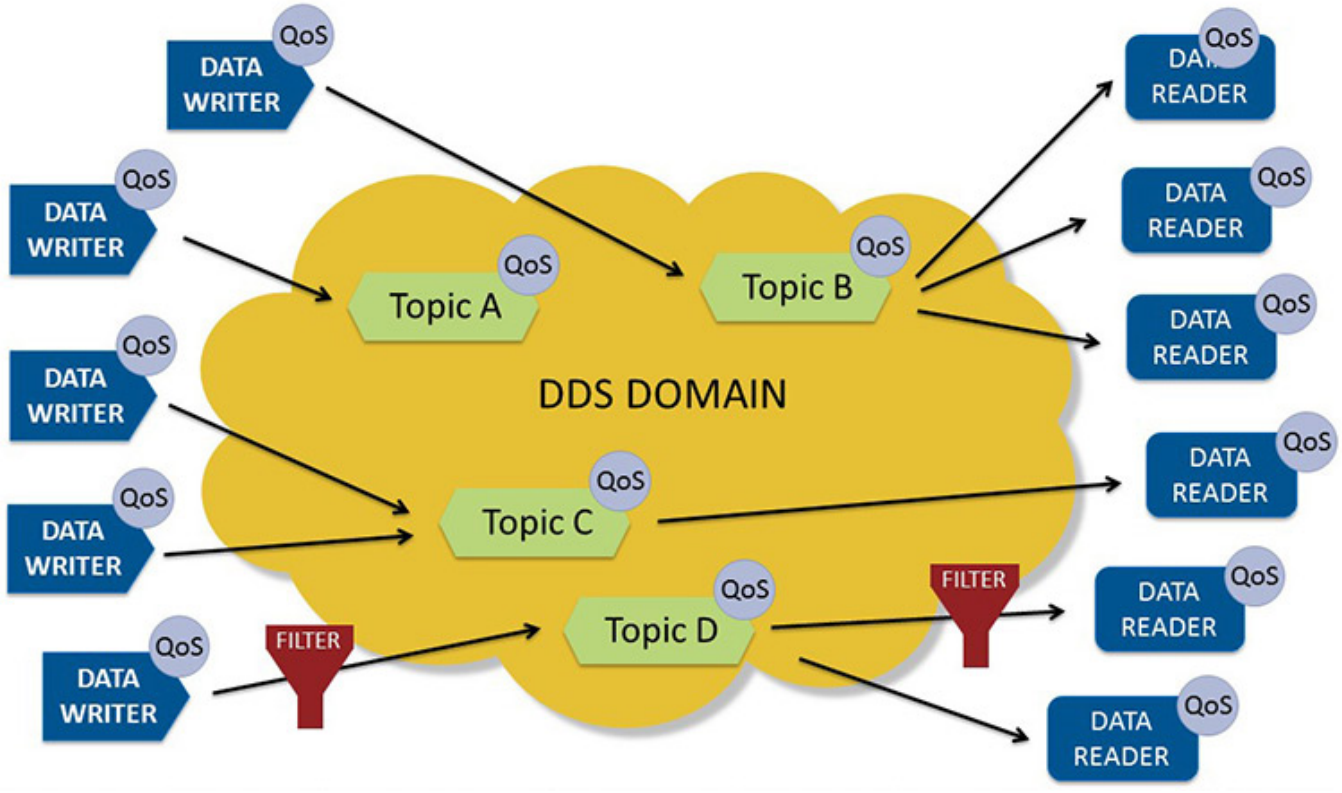
Information Centric Networking (ICN) vs Host Centric Networking (HCN)



- HCN: conversation between hosts — who to talk to.
- ICN: spreads data objects — what to say

relayr.

Data Distribution Service (DDS)



relayr.

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

relayr.

ICN in a nutshell I

- shares packet forwarding with IP mostly
- data consumers and data producers as main entities
- data reliability is a main point
- follows brodcast model: one producer for many consumers

relayr.

ICN request/response outline

- 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

lots 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 ($\neq$ IP)
- data name instead of IP address
- anycasting and multicasting
- consumer can roam — easy mobility

ICN in a nutshell III

- research topic: NSF funded Named Data Networking
- many open questions
- routing
  - congestion control
  - push (event) also, not only polling
- multiple research projects: US, Europe
- watch this space

demo time

relayr.

demo details

- CoAP web client interacting with the coap://californium.eclipse.org:5683 test server
- using Copper ~~Firefox~~ extension with the Basilik browser
- no longer works with Firefox after version 56
- extension needs to be installed from source
- as an alternative browser with a retro look: Pale Moon

conclusions

- there are many type of networks
- the protocol to use depends on what part of which network you are interested in

relayr.

opinionated cheatsheet

- MQTT $\Longrightarrow$ d2c or c2c
- CoAP $\Longrightarrow$ d2d
- HTTP/1.1 $\Longrightarrow$ c2c
- HTTP/2 $\Longrightarrow$ c2c — possibly d2c
- DDS $\Longrightarrow$ c2c & d2c — claims of d2d seem exagerated
- WebSockets $\Longrightarrow$ c2c

bottom line: there is no silver bullet

about me

- GitHub
- LinkedIn
- this presentation: `tinyurl.com/gotoams2017-iot`

relayr.

questions?

relayr.