Internet of Things Protocols - 6LoWPAN and CoAP

Vilen Looga, M.Sc.
Doctoral Student @dps.cse.aalto

About me

- Graduated with NordSecMob M.Sc in 2010.
- Doctoral Student in DPS Group since 2011.
- Research related to energy efficiency and energy modeling.
- Started with smartphones, now focusing on Internet of Things devices.
- Investigate how to do energy management on a network level in IoT.

Outline

- Internet of Things
- Hardware
- Event-driven networking
- 6LoWPAN at a glance
- CoAP in depth
- Messages
- Observe
- Conclusions
- References

50 billion connected devices

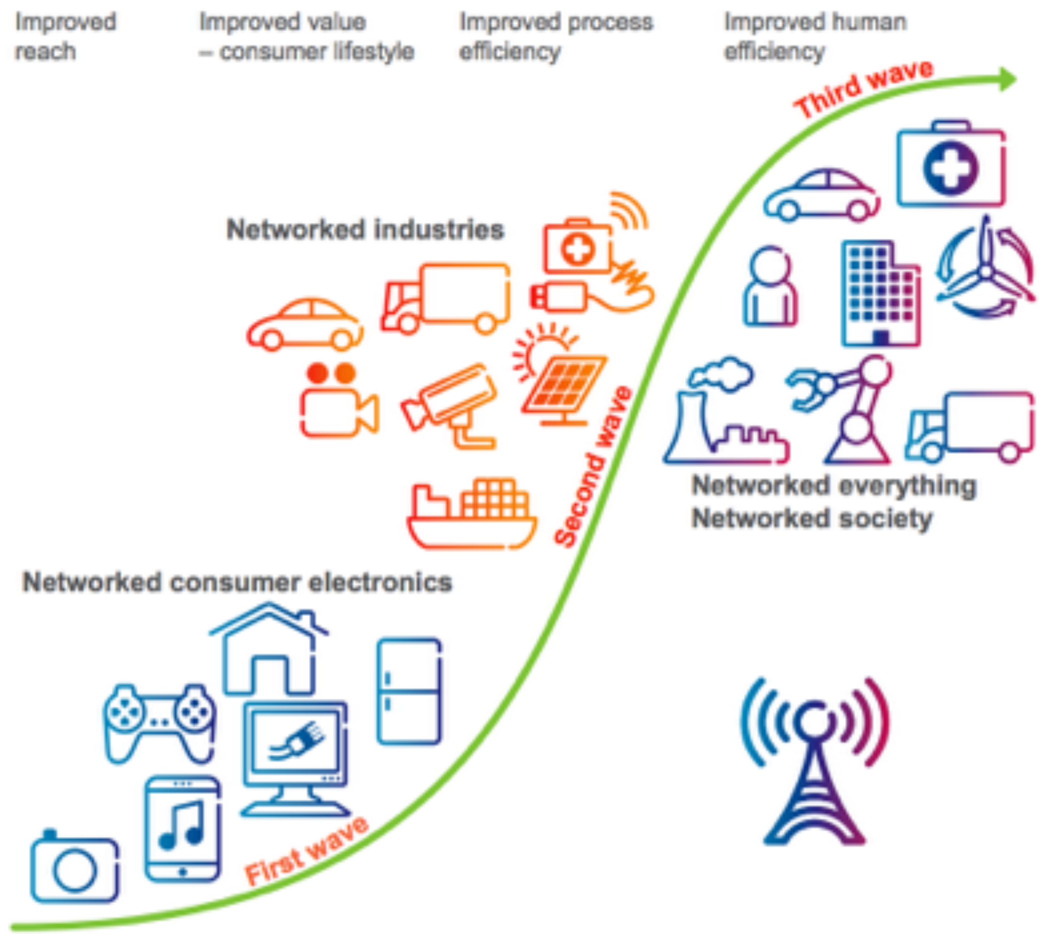
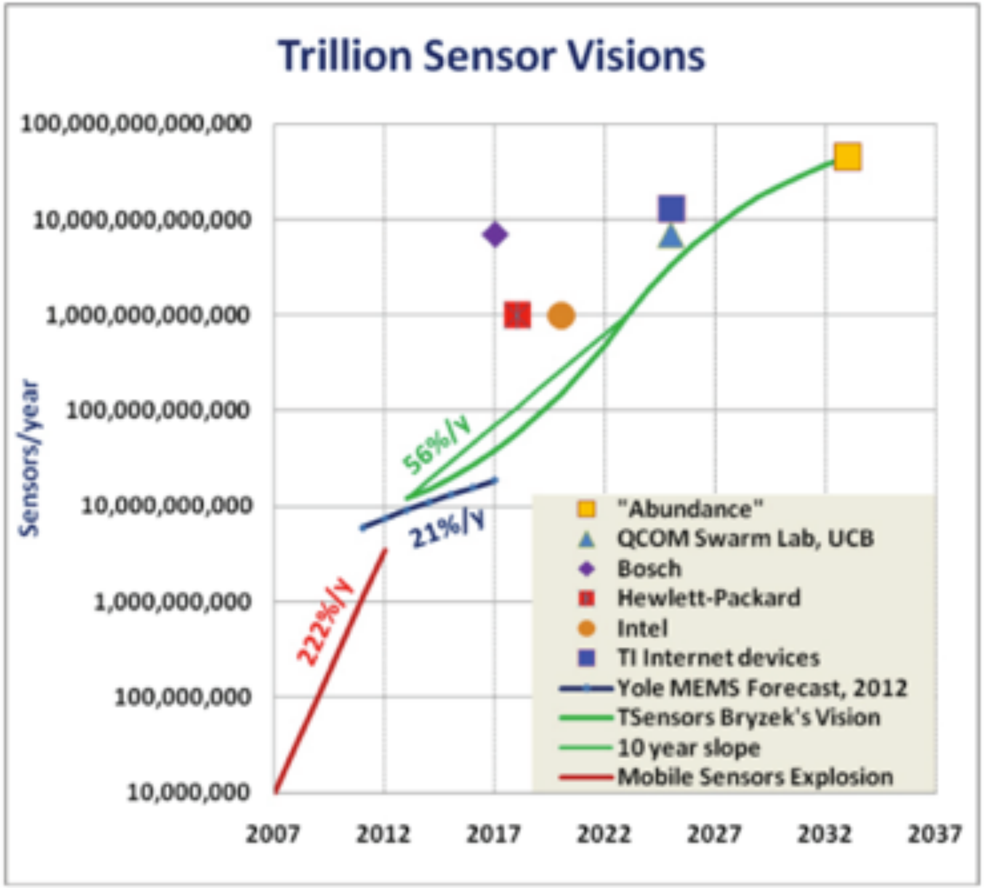


Image by Ericsson©

1 Trillion Sensors (yes, with a "T")



1 Trillion Sensors (yes, with a "T")

Consumer IoT

Image by NXP Semiconductors©

Industry IoT - Machine to machine (M2M)

Image by Schneider Electric©

Enevo ONe

Enevo ONe is a complete waste monitoring solution that brings up to 50% savings in waste collection costs. Works with **any type of container** and **any type of waste** mixed, glass, bio, metals or fluids such as oils and waste water etc.

What's included

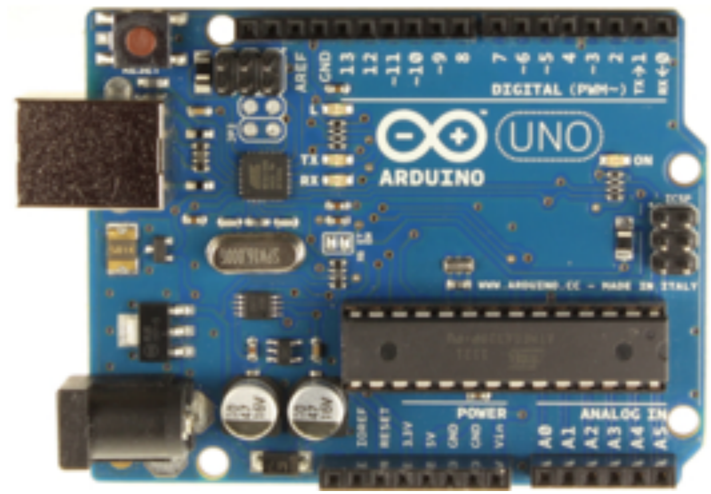
- Sensor Device
- Access to the ONe Collect Web Portal
- Real Time Fill-Up Measurements
- Collection Forecasts
- Daily Collection Lists
- Alert Service
- Sensor Warranty and Replacement
- All Data Transmissions
- Unlimited Number Of User Accounts
- 24/7 Email Support

Aalto University
School of Science

Image by Enevo©

Node hardware - previous gen.

Zolertia Z1 (Contiki OS)

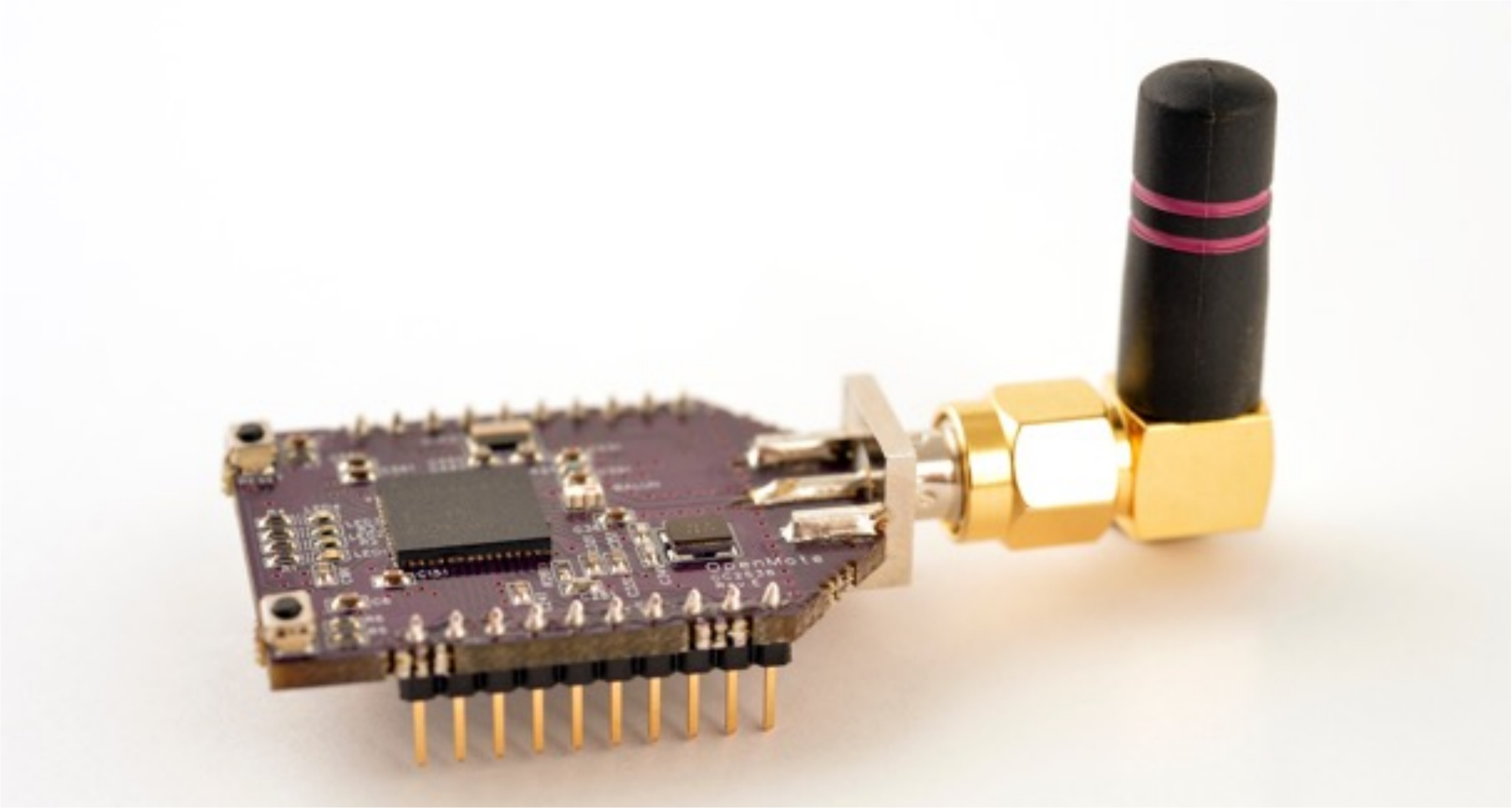


Arduino

Node hardware - previous gen.

- Limited resources:
  - 8KB RAM, 92KB ROM, 16-bit CPU
- Wireless communication: 6lowPAN over 802.15.4, GPRS, 4G
- Long-lasting battery

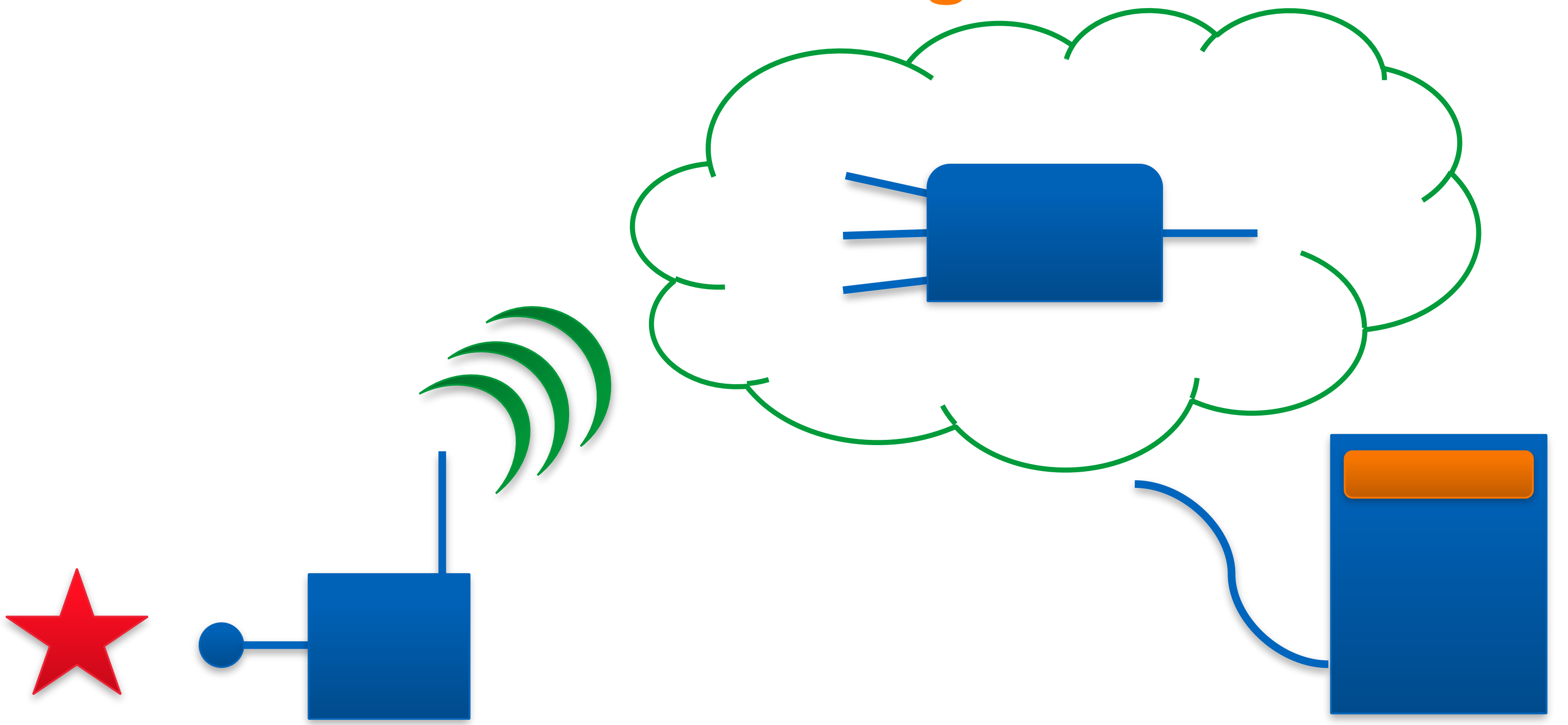
Node hardware - current gen.



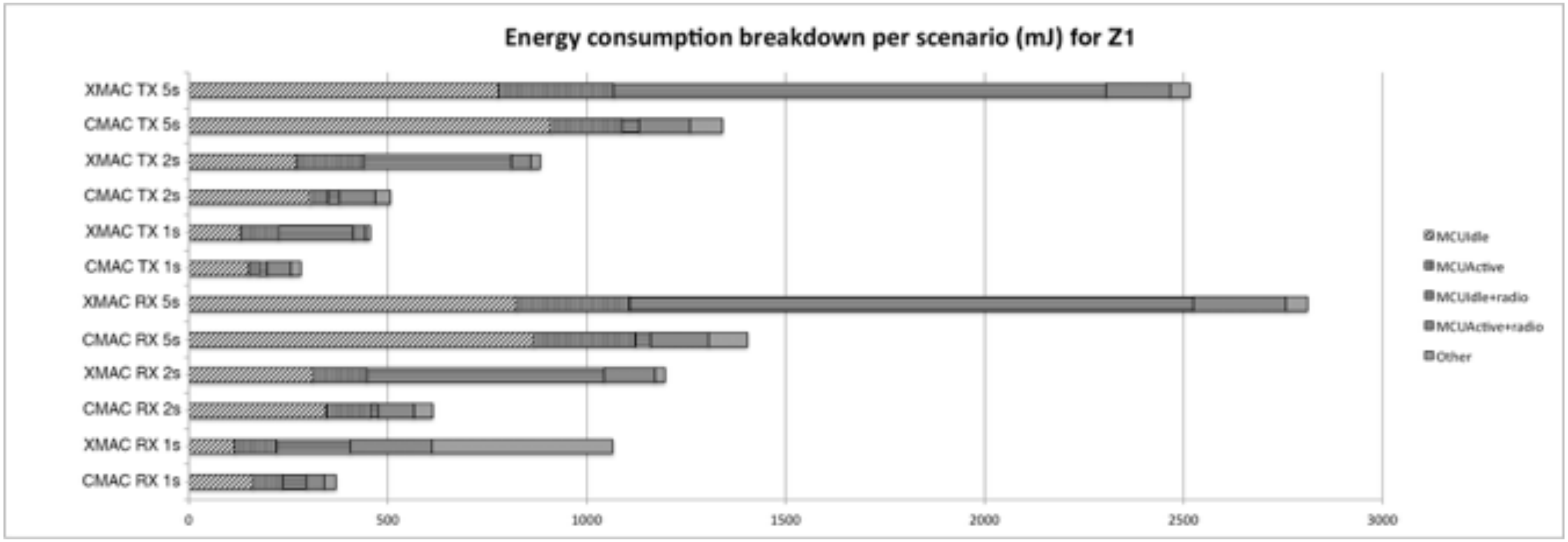
Node hardware - current gen.

- Still a bit limited:
  - 32 MHz 32-bit Cortex-M3 ARM CPU
  - 32 KB RAM, 512 KB Flash
- Connectivity:
  - 6LoWPAN over 802.15.4, 4G, Low-Power WiFi

Event-driven networking



Radio dictates energy consumption



IoT Network Stack

TCP/IP Protocol stack	IoT protocol stack
HTTP	CoAP
TCP, UDP	UDP
IPv4, IPv6	6LoWPAN
Ethernet MAC	ContikiMAC, X-MAC, TSCH
Ethernet PHY	IEEE 802.15.4 PHY, Low-power Wifi, 4G

IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)

Features

- "The Internet protocol should be applied to the smallest low-power devices with limited processing capabilities" - *Shelby & Bormann*

- Stateless header compression
- Network autoconfiguration using neighbour discovery
- Support for 802.15.4:
    - addressing schemes (16-bit short address within the PAN)
    - fragmentation

Network autoconfiguration

- Link-layer connectivity between nodes(commissioning):
  - Channel, modulation, data rate
  - Addressing mode (64- or 16-bits)
  - MAC mode (ContikiMAC, TSCH)
- Network layer address configuration, discovery of neighbors, registrations (bootstrapping)
- Routing algorithm sets up paths (route initialization)
- Continuous maintenance of

802.15.4 support: fragmentation

- MTU size: 802.15.4 leaves only ~100 bytes per frame for payload, while max MTU size in IPV6 can be 1280 bytes
- Fragmentation is done by assigning tags to IP packets at the source of fragmentation
- The payloads are assembled at the destination once all the fragments are received (order of arrival is not important)

Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP)

- Application level protocol over UDP
- Designed to be used with constrained nodes and lossy networks
- Designed for M2M applications, such as home and infrastructure monitoring
- Built-in resource discovery and observation (“push notification”)
- Block-wise transfer

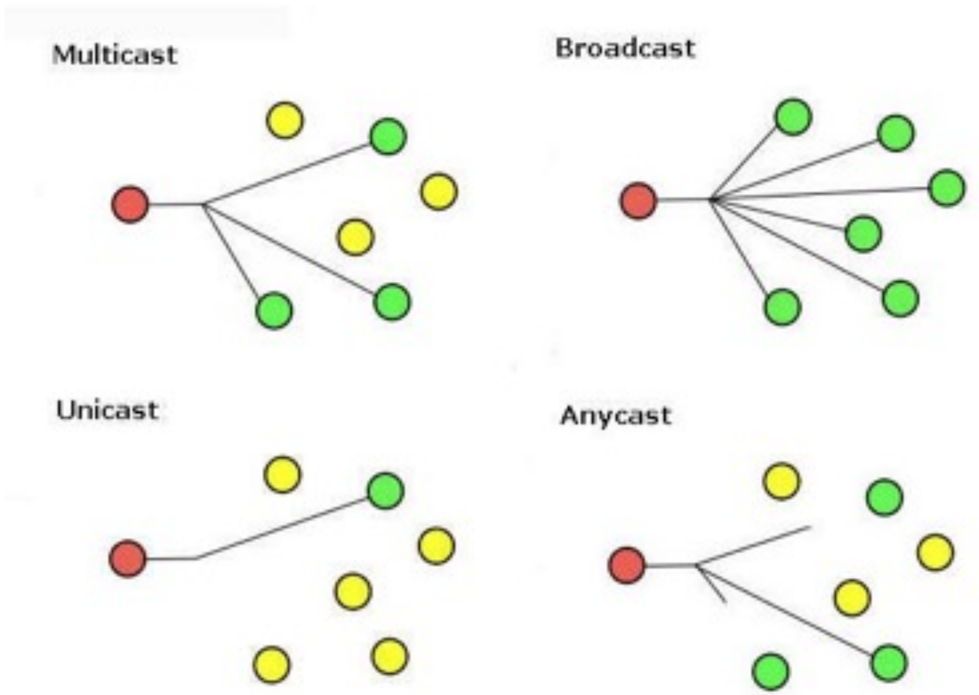
Constrained Application Protocol (CoAP)

- RESTful for easy interfacing with HTTP
- Multicast support
- Low overhead and simple
- Should fit into a single UDP packet or IEEE 802.15.4 frame

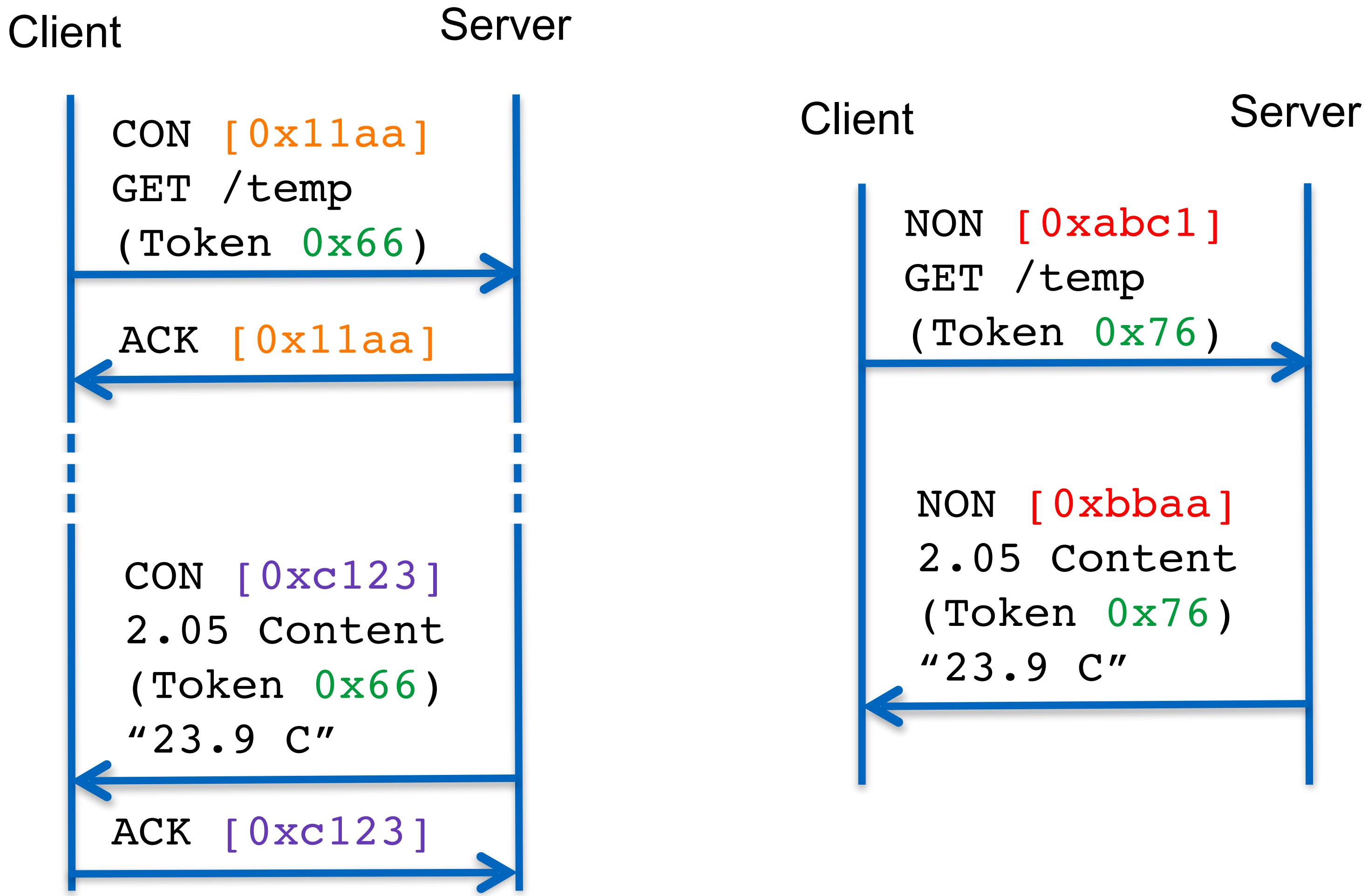
(Helper slide) RESTful

- REpresentational State Transfer is a software architectural style for Web client-server
- Resources are represented as URL:
  - “example.com/profile/johnny”
  - “example.com/domain/sensor3/temp1“
- Resources can be retrieved and manipulated using VERBS:
  - GET, POST, PUT, DELETE
- Example protocol: HTTP

(Helper slide) Uni-, multi-, broadcast



Request/Response model



CoAP message

UDP packet

SRC port	DST port	Length	CRC	CoaP MSG

CoAP message

Ver	T	OC	Code	Message ID
Option Delta			*Length*	*Value*
		...		
Payload				

CoAP message

42	01	02:4c	9b	2e:77:65:6c:6c:2d:6b:6e:6f:77:6e	04	63:6f:72:65

01	*00*	*0010*	*00000001*		*0001*	*1011*		*0000*	*0100*	
Ver. 1	*Type 0*	*Options 2*	*Request*	*Message ID*	*0+9=9: Uri-Path*	*Length 11*	*".well-known"*	*9+0=9: Uri-*	*Length 4*	*"core "*

Message types

- Confirmable (0)
- Non-Confirmable (1)
- Acknowledgement (2)
- Reset (3)

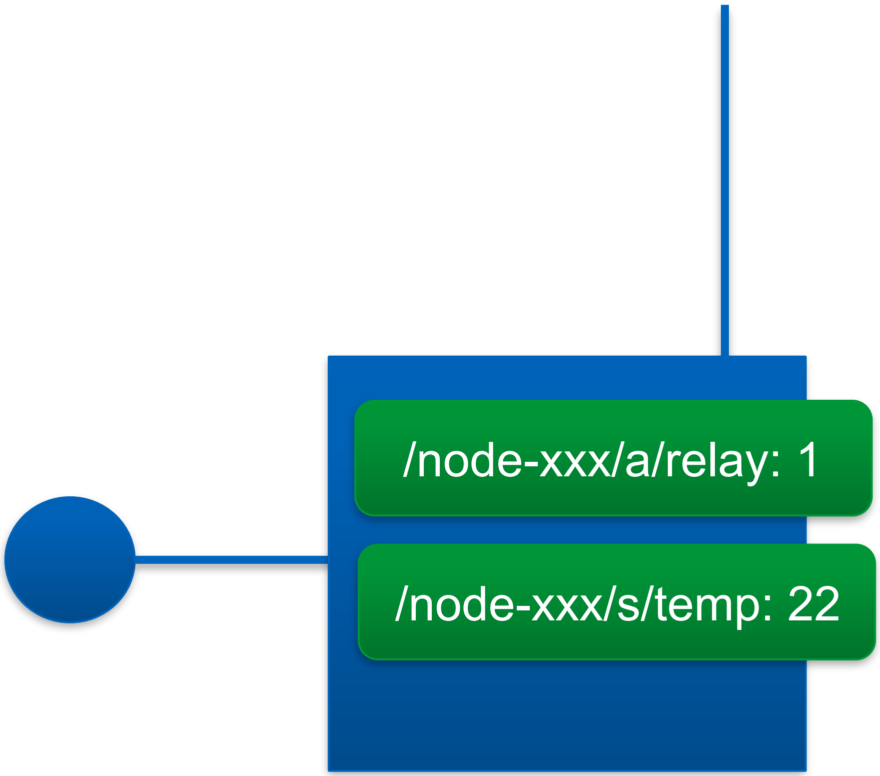
Message Code

- Request (1-31)
- Response (64-191)
- Empty (0)

Options

- 1 If-Match
- 3 Uri-Host
- 4 ETag
- 5 If-None-Match
- 7 Uri-Port
- 8 Location-Path
- 11 Uri-Path
- 12 Content-Format
- 14 Max-Age
- 15 Uri-Query
- 17 Accept
- 20 Location-Query
- 35 Proxy-Uri
- 39 Proxy-Scheme
- 60 Size1

RESTful protocol



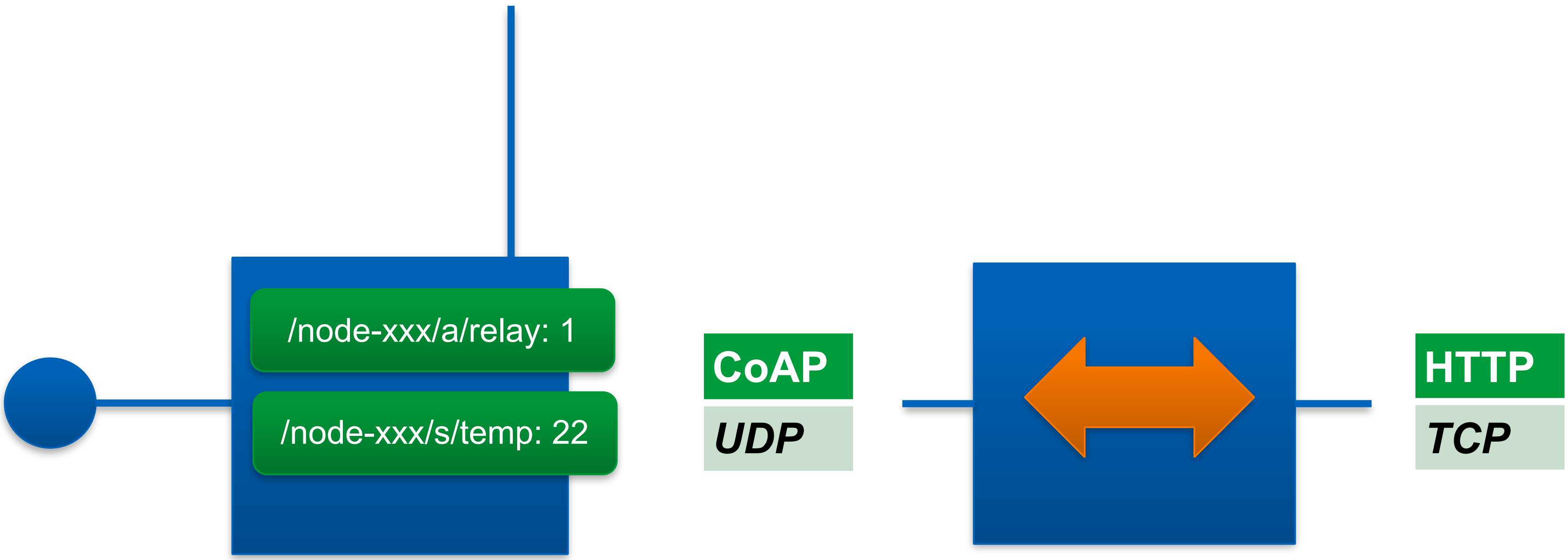
GET /node-xxx/a/relay
-> 1

GET /node-xxx/s/temp
-> 22

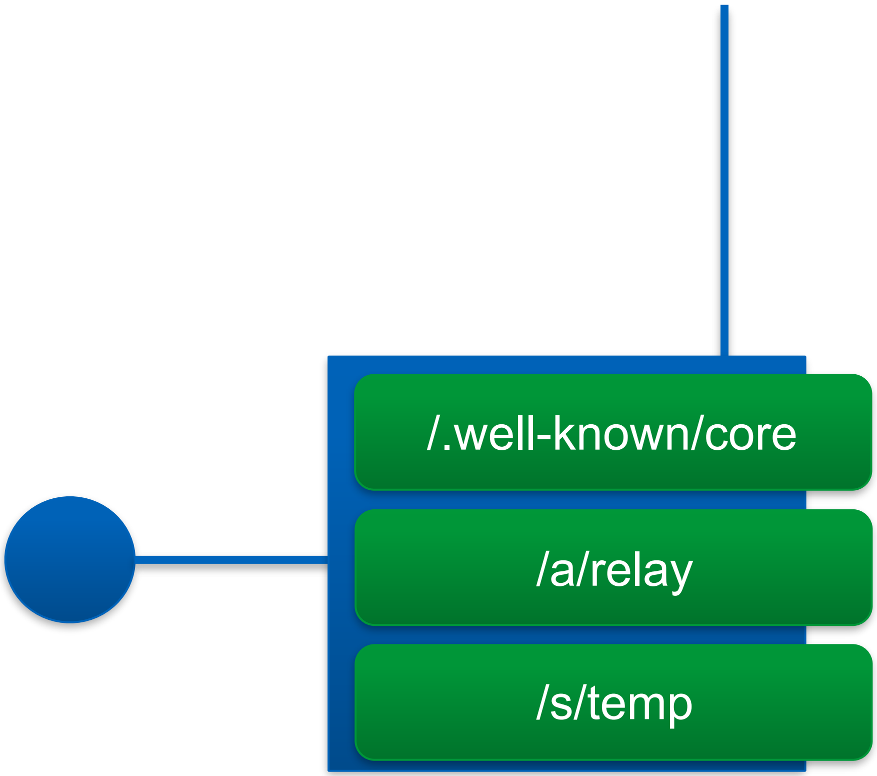
POST /node-xxx/a/relay/1
-> OK

DELETE /node-xxx/a/relay
-> OK

CoAP to HTTP proxy



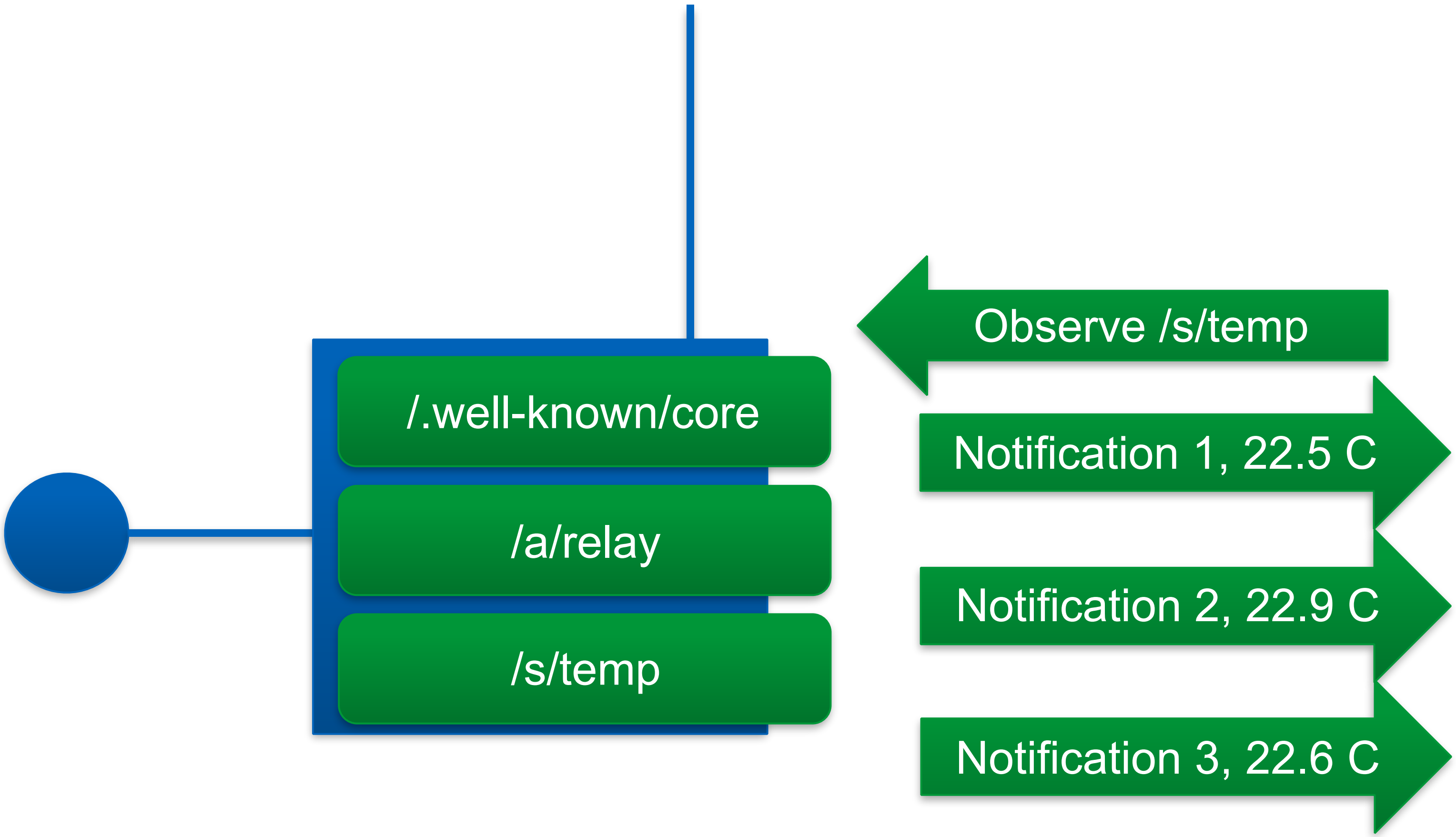
Resource discovery



GET /.well-known/core
->

</a/relay>;rt=”Relay";ct=0,
</s/temp>;rt="Temperature";ct=0

Observe functionality



Multicast

- One-to-many has obvious application in IoT networks:
  - Data and code dissemination
  - Delivering event data to a set of interested servers
  - Measurement data to more than one subscriber
- Current multicast proposals have drawbacks:
  - SMRF: one-way directional flooding; only useful for data flowing from border router to nodes
  - PIM-WSN: too many forwarding states inherited from PIM-SSM alleviated by Bloom filters

Block-wise transfer

- Transferring large resources with minimal overhead
- Avoid IP fragmentation (datagram > MTU)
- No need to manage state at IP or adaptation layer
- Both parties agree on block size

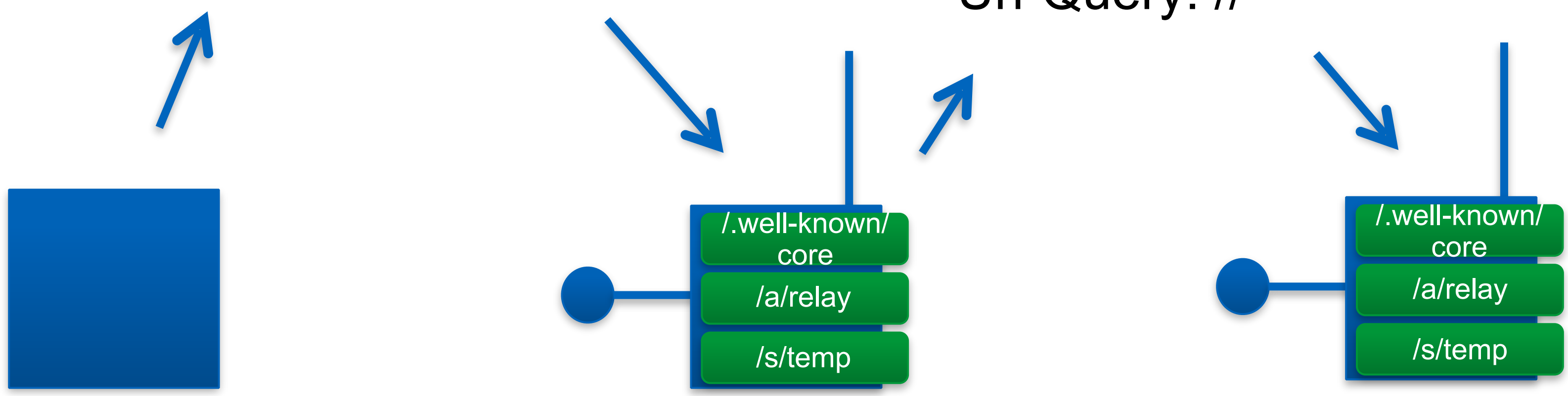
Multicast

- Our proposal (Deng Y., Doctoral student):
  - Any node can be the sender
  - Interoperability with outside domains
  - Link layer broadcast
  - One network interface per node
- RPL (Routing protocol for Low-power and Lossy networks):
  - Uses tree structure
  - Multicast messages can be sent in RPL control messages

Proxy

Proxy-URI: "coap://domain.fi:
60001/node-00x/s/temp/"

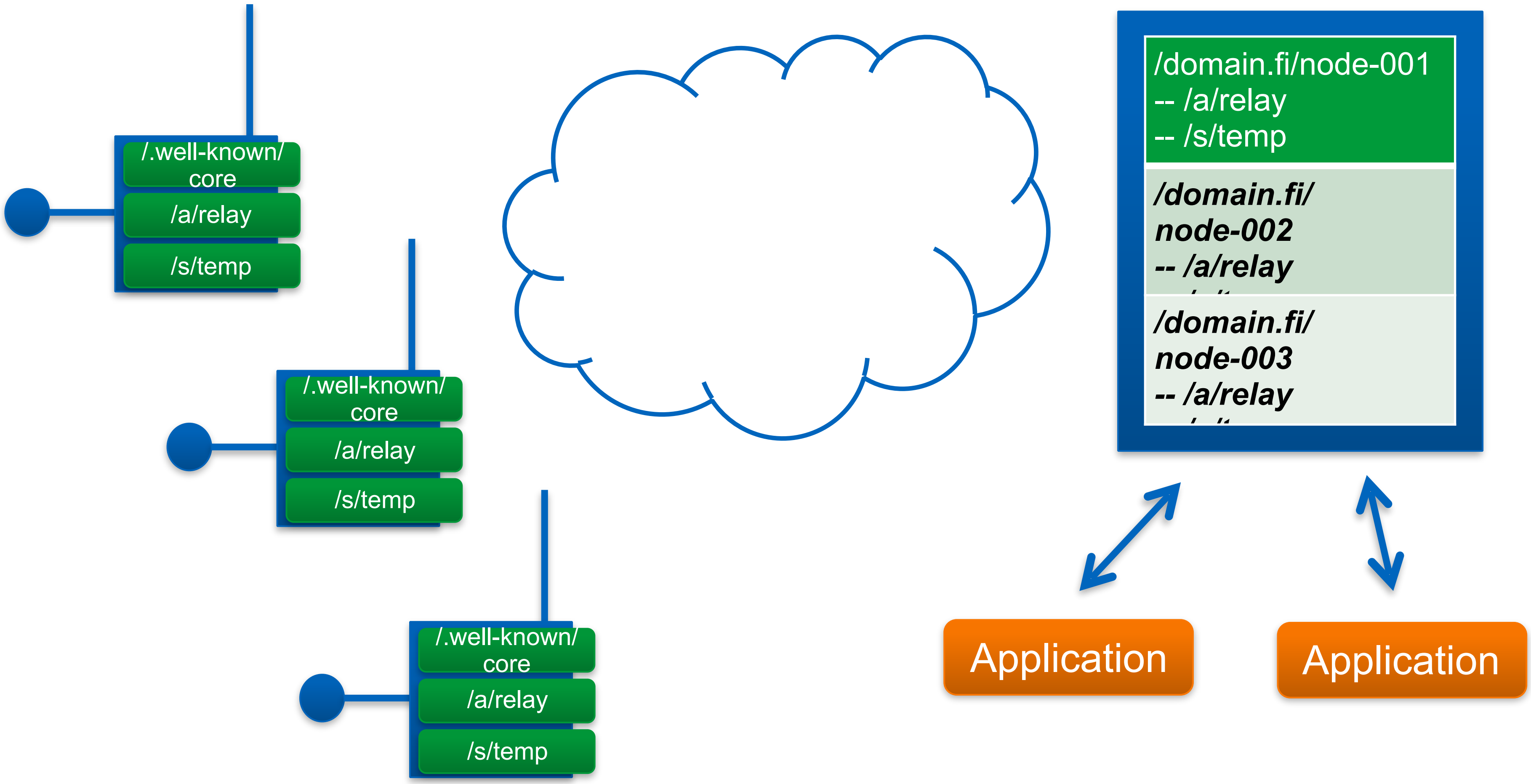
Uri-Host: domain.fi
Uri-Port: 60001
Uri-Path: s
Uri-Path: temp
Uri-Query: //



Security

- Similarly to TLS used in HTTP (-> HTTPS), CoAP is secured using Datagram TLS (DTLS)
- DTLS = TLS + features to deal with unreliability of TLS:
  - DTLS records are independent: if record N is lost N+1 can still be decrypted, while N is retransmitted
  - TLS handshake breaks if the packets are out of order; DTLS queues handshake messages until the correct one
  - Application is responsible for dealing with packet reordering, loss, data re-assembly etc.
- Minimal implementation of MUST configurations:
  - Not all cipher suits supported
  - DTLS does not work for multicast communication
- Devices should keep connection open as long as possible to avoid mutual authentication setup overhead

Resource directory



P2P Resource Directory

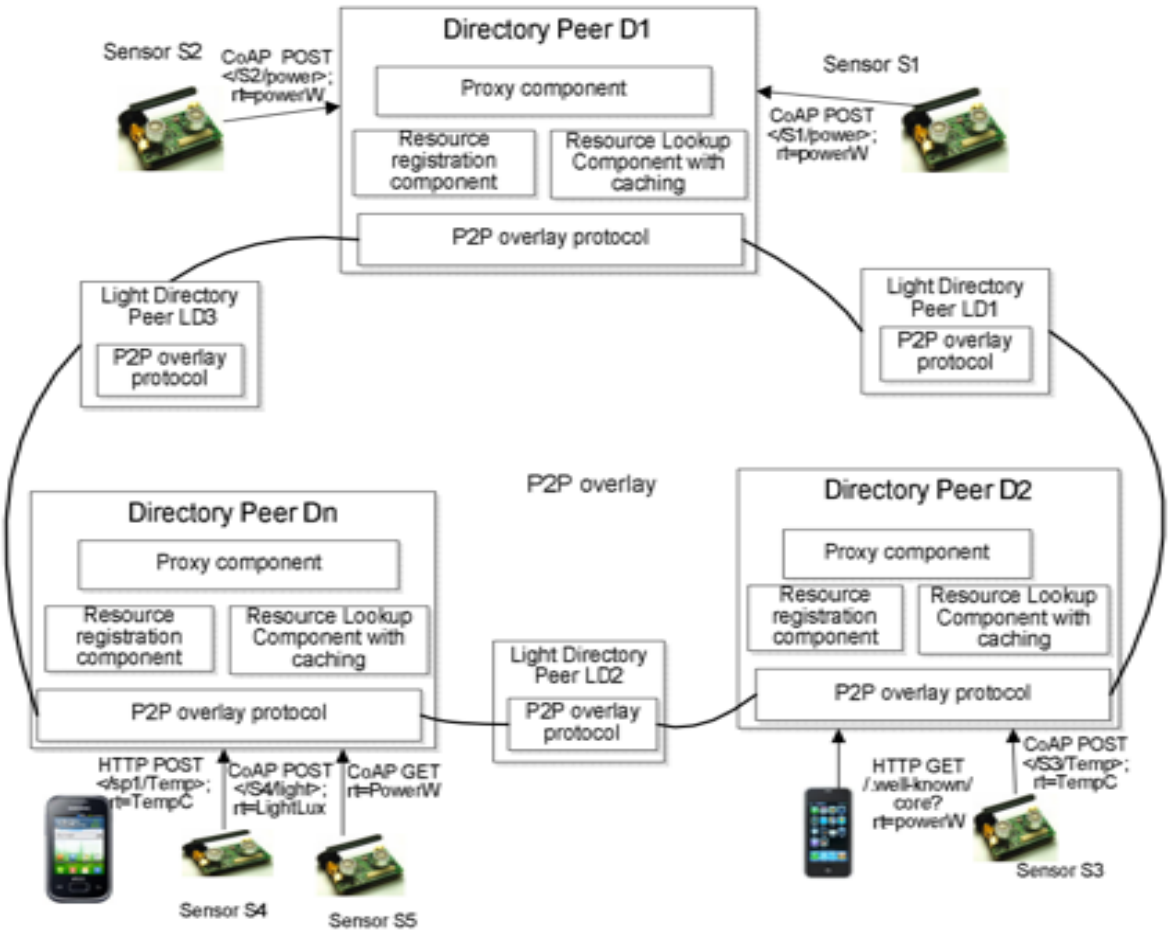


Image by Meirong Liu

Hardware

Tiny CoAP sensor by Ericsson Research

- Ethernet, IPv6, UDP
- 48 lines of assembler code
- Does not support Observe

Image by Ericsson©

Implementation

Software	Type	Supported Platform
libcoap	*Client, Server*	*x86, Contiki, tinyOS*
Californium	*Client, Server*	*Java*
Copper	*Client*	*Javascript*
Erbium	*Client, Server*	*Contiki*
… and many many more		

Future/baseless speculation

- CoAP will most likely peak at some point and give way to HTTP even on the most low-powered nodes
- Reasons:
  - Nodes are becoming powerful enough (see OpenMote)
  - Battery life improves
  - 802.15.4 replaced by WiFi/4G/5G
  - Industry wants to have IPv6/TCP/HTTP end-to-end
- Lesson: understand the underlying problem, don't get too attached to a solution

Energy management in large scale IoT networks (Looga et al.)

- Problem:
  - The potential for energy savings is nearing it’s end when it comes to the physical layer
  - Large-scale IoT networks require different kind of energy management of nodes, that should be scalable and with minimal overhead
- Assumptions:
  - Network traffic from M2M nodes is rather predictable
  - Network traffic determines the energy consumption of the node
  - Network topology in production systems is simple

Energy management in large scale IoT networks (Looga et al.)

- Our proposal: Traffic-based energy model for 802.15.4 nodes
  - requires only the traffic traces from the node and some of the nodes configuration parameters (MAC layer type, TX/RX values etc.)
  - No overhead on the node
  - Scales linearly with the number of nodes
  - Has good precision (>90%)

References

- 6LoWPAN: http://datatracker.ietf.org/wg/6lowpan/documents/
- IETF CoAP draft: https://datatracker.ietf.org/doc/draft-ietf-core-coap/
- IETF CoAP Observe draft: http://tools.ietf.org/html/draft-ietf-core-observe
- Jari Arikko. Tiny CoAP Sensors. http://www.arkko.com/publications/ietf81_lwip_tiny.pdf
- DTLS: http://tools.ietf.org/html/rfc4347
- More than 50 billion connected devices – taking connected devices to mass market and profitability (whitepaper). Ericsson
- Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011–2016. Cisco

Thank you!

Questions?