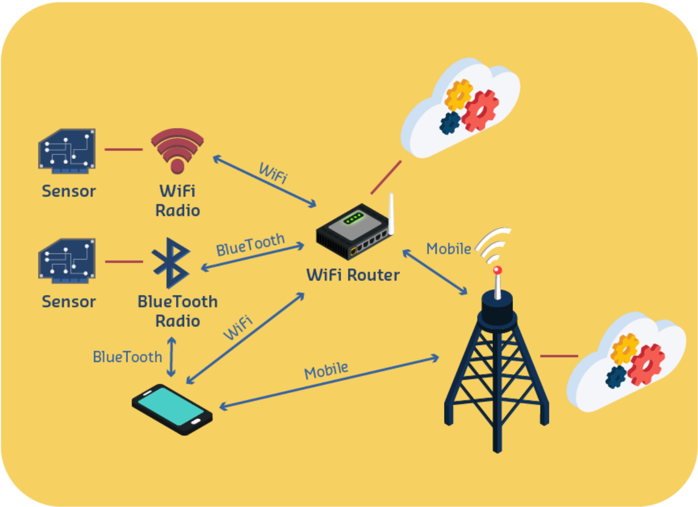
Internet of Things and Services

Service-oriented architectures

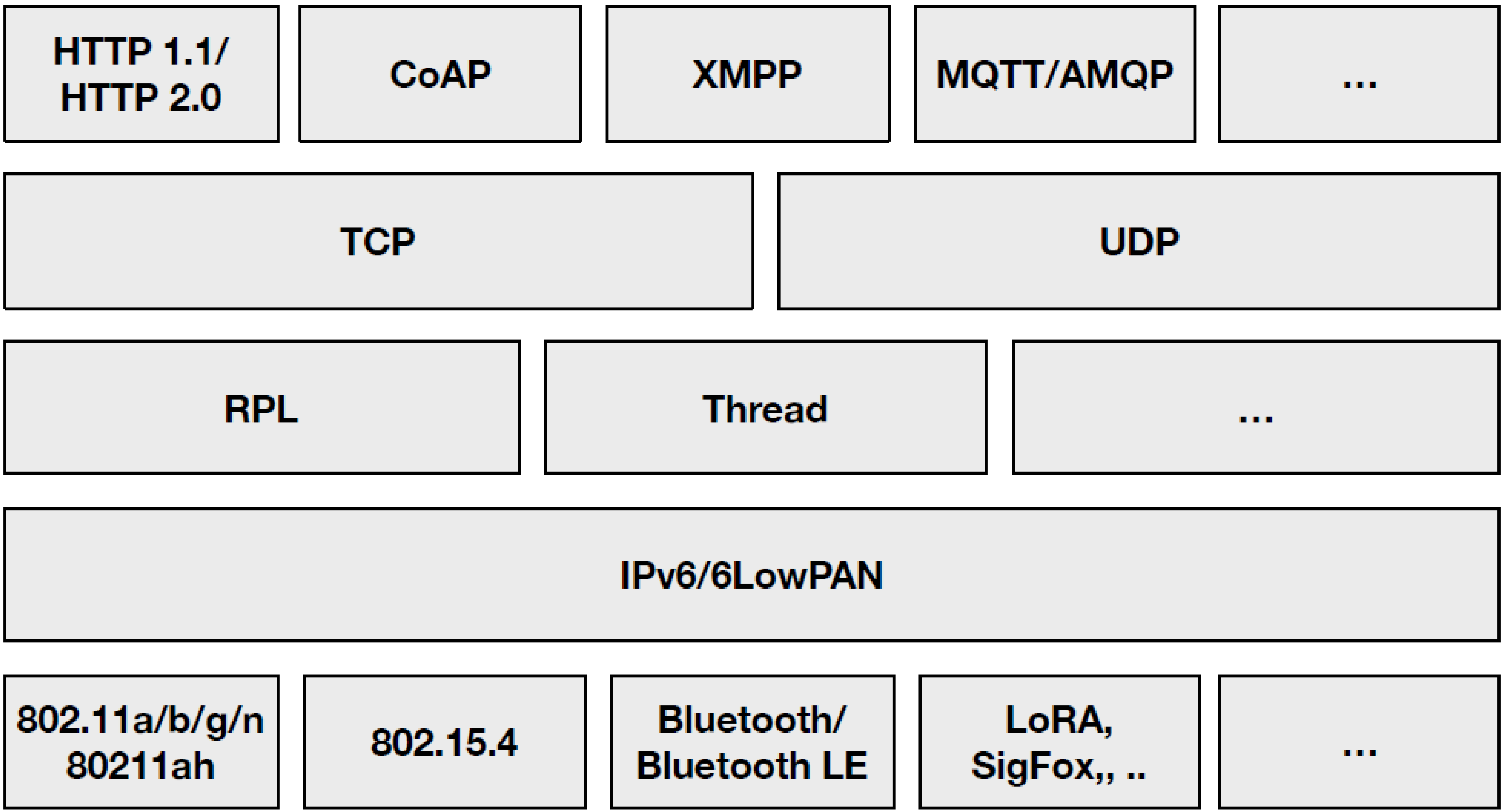
IoT networks & protocols

Department of Computer Science
Faculty of Electronic Engineering, University of Niš

IoT communication architecture



IoT protocol stack



More IoT communication protocols

- Standards for Industrial IoT, connected vehicles, PLCs,…

PROFINET

PROFIBUS

The nice thing about standards is that you have so many to choose from

[Andrew S. Tanenbaum]

Modbus

HART COMMUNICATION PROTOCOL

CAN

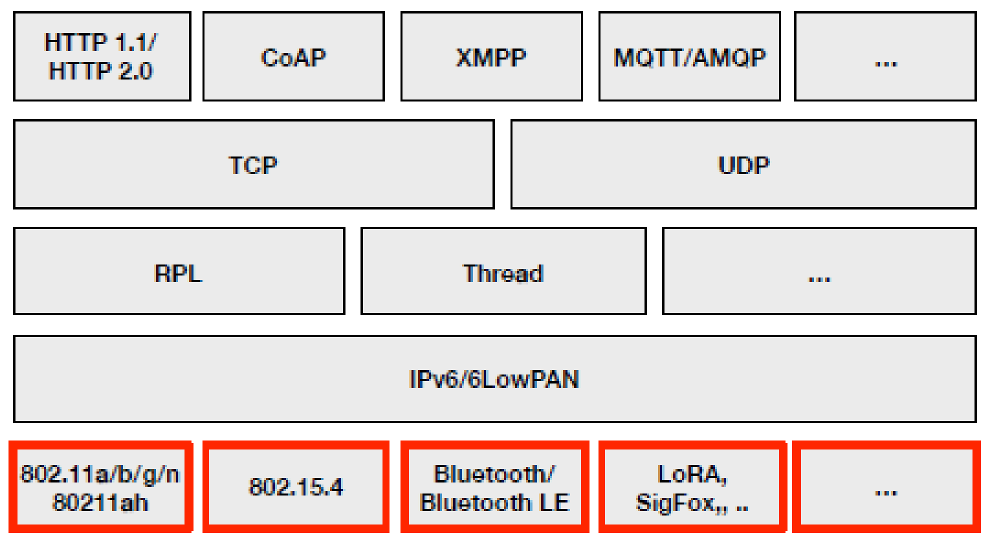
EtherCAT

Fieldbus Foundation

And many more

IoT Physical & Data link layer

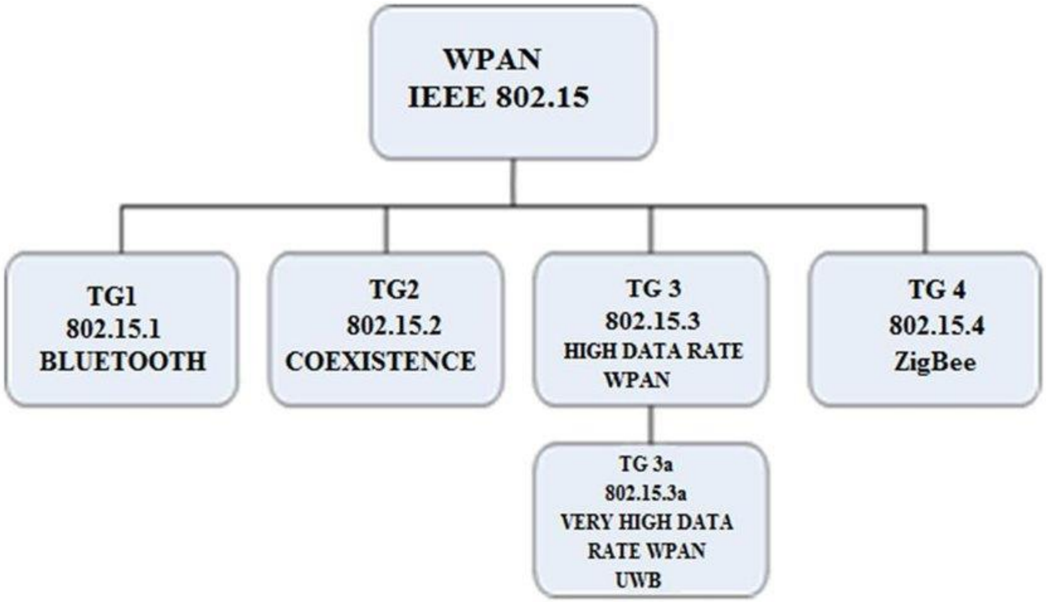
- Essentially a melting pot
- Only common characteristic
  - Reducing energy consumption
- Many different complementary goals
  - Long range
  - Ease of integration
  - Backward compatibility
  - ....



Non-IP based WPAN

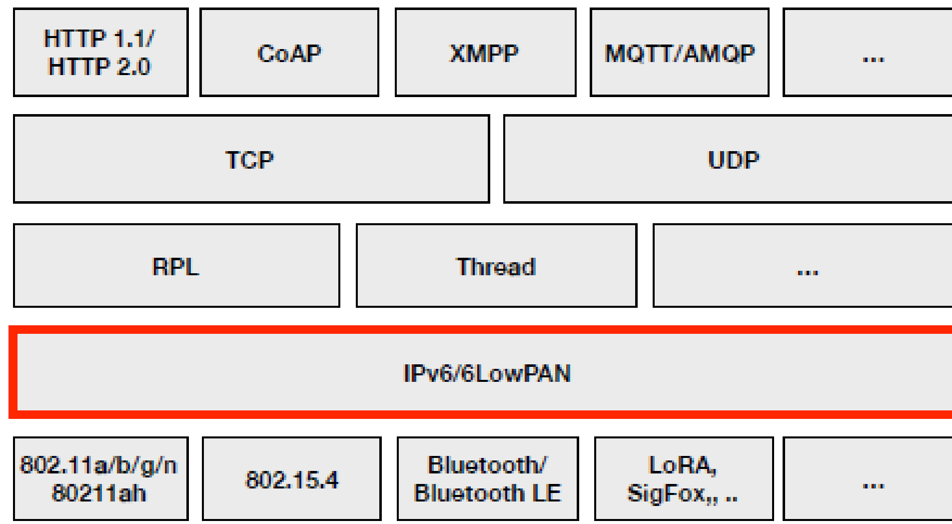
- IEEE 802.15 Standards

- Bluetooth 802.15.1
- Bluetooth Low Energy – BLE (ver 5.0)
- ZigBee - 802.15.4
- Z-Wave



IoT Network layer

- Internet connectivity - IPv6
- Standard IPv6 is too heavyweight
  - 6LowPAN provides
    - Fragmentation
    - Encapsulation
    - Header compression
- Use intermediate gateways
  - Application-specific addressing within the IoT network



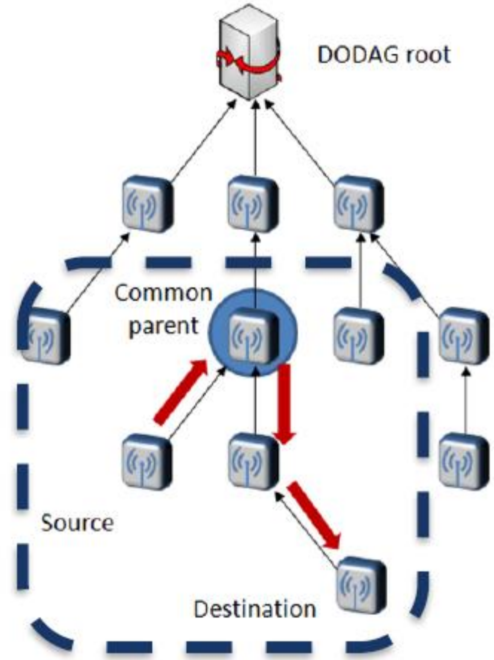
IP based WPAN & WLAN

- WPAN with IP – 6LoWPAN
  - An adaptation layer for IP6
  - The protocol can be used with other WPAN protocols, such as 802.15.4 and Bluetooth

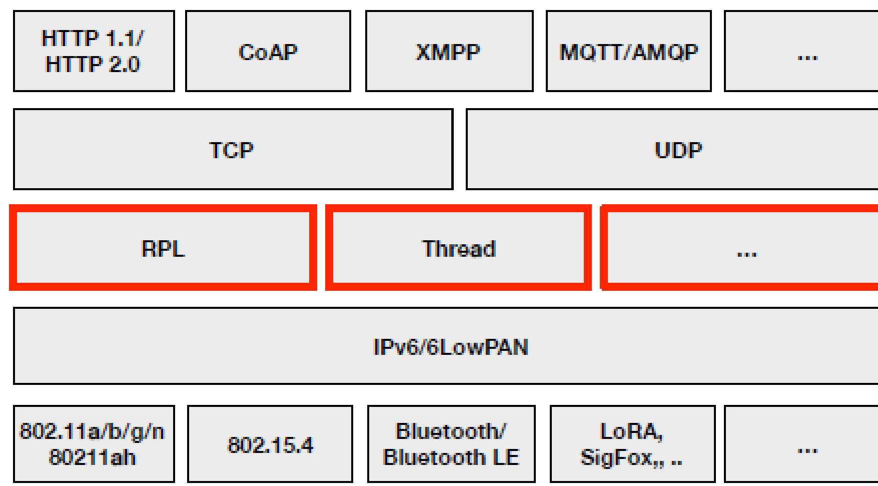
6LoWPAN Protocol Stack	Simplified OSI Model
HTTP, CoAP, MQTT, Etc.	5. Application Layer
UDP, TCP Security: TLS/DTLS	4. Transport Layer
IPV6, RPL 6LoWPAN	3. Network Layer
IEEE 802.15.4 MAC Layer	2. Data Link Layer
IEEE 802.15.4 PHY	1. Physical Layer

- Thread
  - Based on 6LoWPAN and 802.15.4 with IP
  - Thread Group Alliance (2014): Alphabet (Google holding company), Qualcomm, Samsung, ARM, Silicon Labs, Yale and Tyco.
- IEEE 802.11 suite of protocols
  - a, b, g, n, ac, ah, p, af, ad, ax

IoT Routing

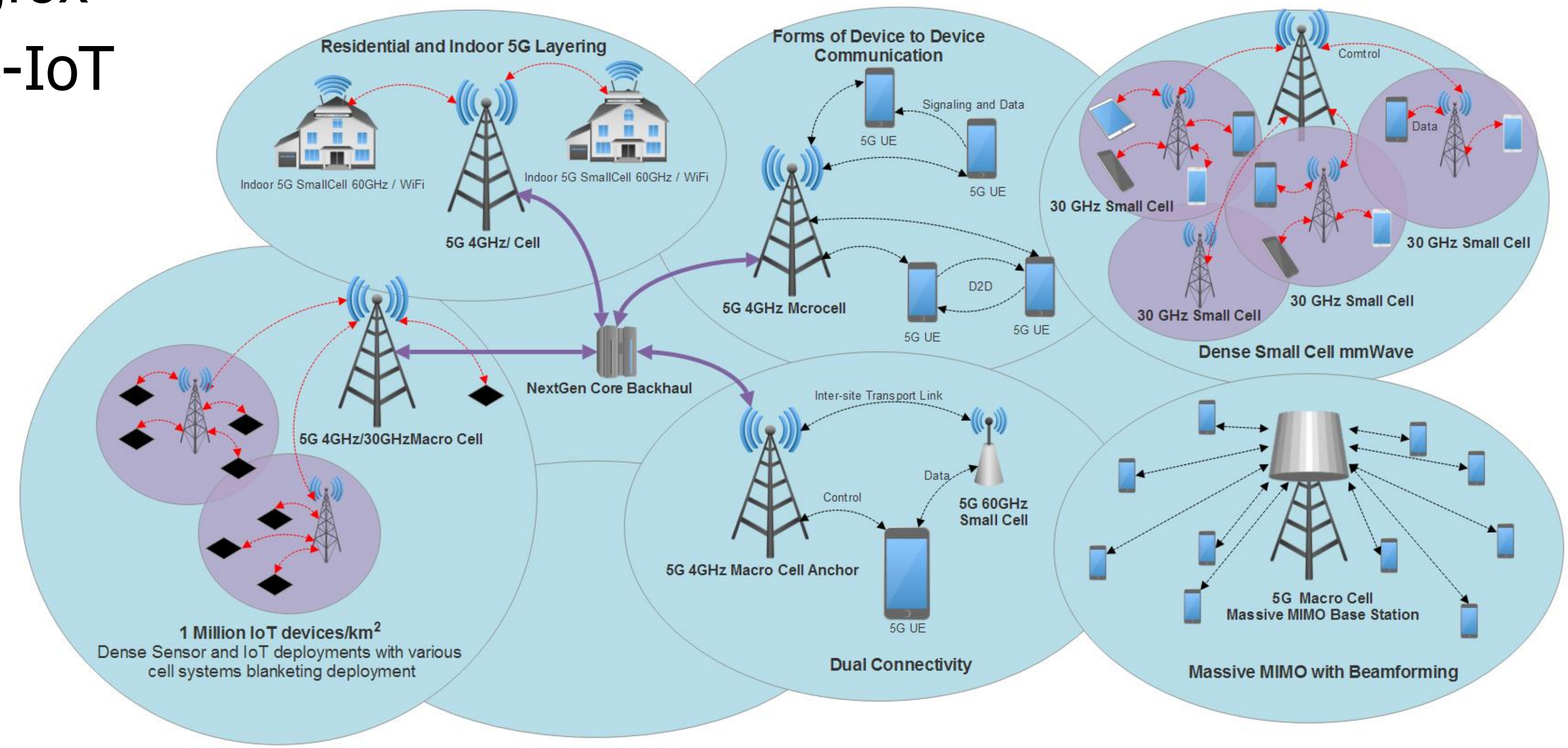


- Adaptation layer
- Mesh networking
- Many-to-many traffic
  - Yet, the bulk is to/from the Internet
- RPL - Routing Protocol for Low-Power and Lossy Networks
  - Directed Acyclic Graph (DODAG)
  - Open specification, 802.15.4
- Thread
  - Closed
  - Home automation
- Many others…
  - Also, for non-IP networks



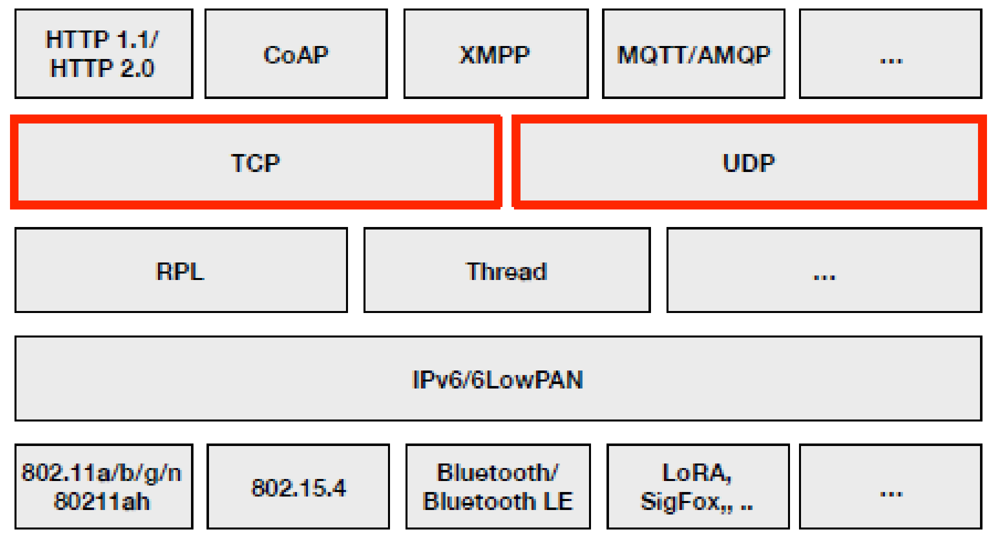
Low Power WAN

- 5G
- LoRa and LoRaWAN
- Sigfox
- NB-IoT



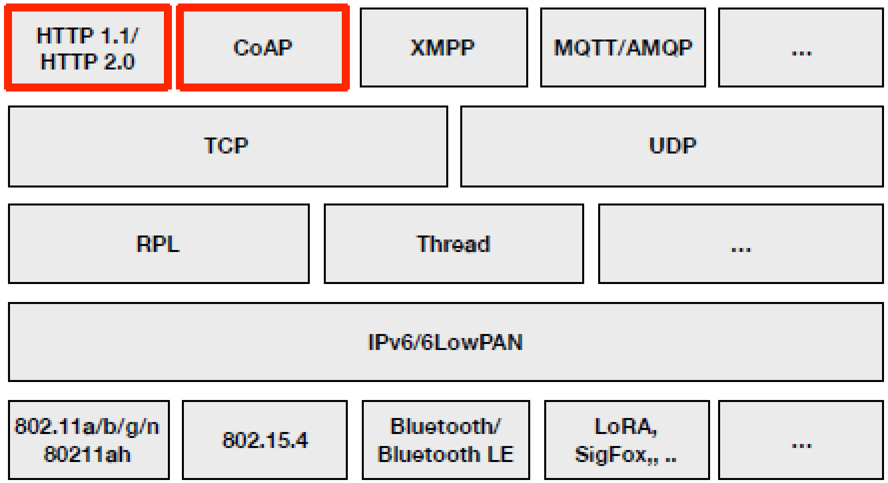
IoT Transport layer

- TCP extremely costly
  - Especially multi-hop
  - Congestion control not designed for wireless
- Use UDP and build reliability on top
  - If and when needed



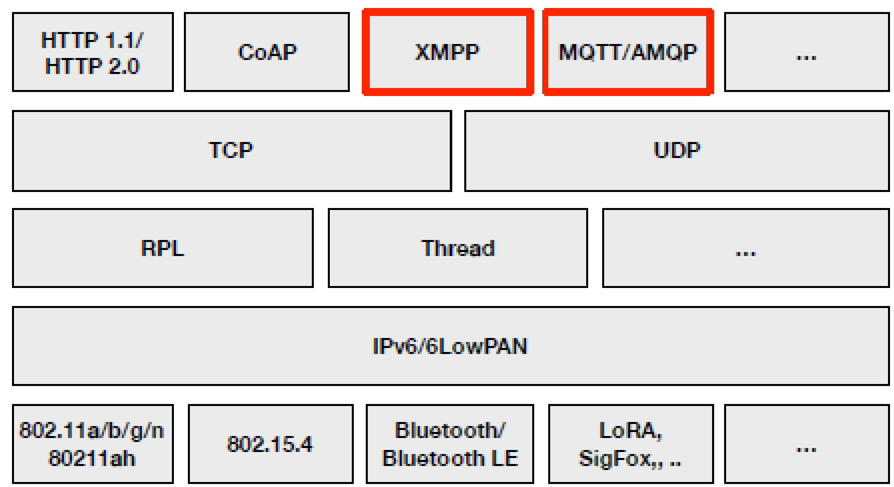
IoT Application layer

- Request/Response plus REST
- HTTP is very heavyweight
  - Text protocol
  - Difficult to push to clients
- CoAP (IETF)
  - Binary, compact
  - Uses UDP, reliability on top
  - Observers
- HTTP/2 (HTTP 2.0)



IoT Application layer - Messaging

- Message oriented
  - Publish/Subscribe
  - Brokers and end-points
- MQTT (IBM, now open source)
  - Lightweight and simple
  - QoS over TCP
  - Hierarchical topics, payload agnostic
- AMQP (Microsoft)
  - Same QoS as MQTT
  - Route with meta-data and topics
- XMPP, STOMP, …



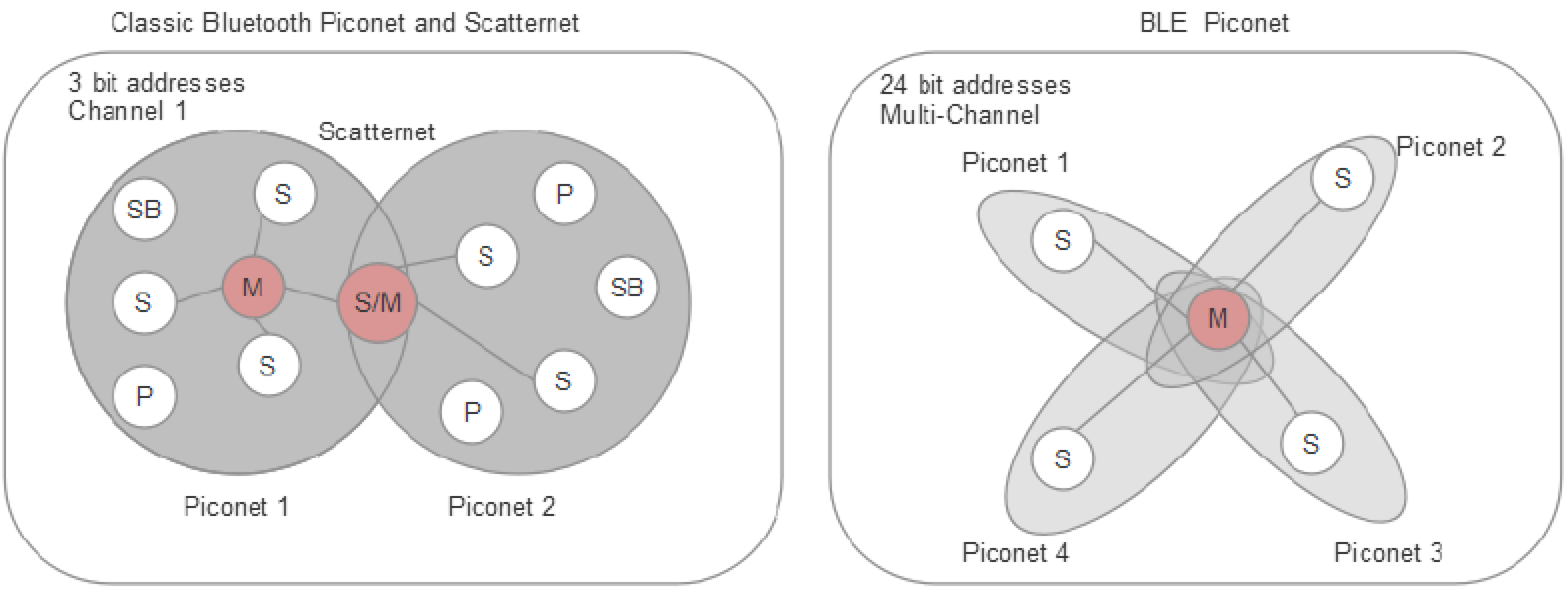
INFRASTRUCTURE PROTOCOLS

Bluetooth vs. Bluetooth LE

Technical Specification	Classic Bluetooth	Bluetooth low energy
Distance/Range (theoretical max.)	100 m (330 ft)	50 m (160 ft)
Over the air data rate	1–3 Mbit/s	1 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Security	56/128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check
Latency (from a non-connected state)	Typically 100 ms	6 ms
Total time to send data (det.battery life)	100 ms	3 ms , <3 ms
Voice capable	Yes	No
Network topology	Scatternet	Scatternet
Power consumption	1 as the reference	0.01 to 0.5 (depending on use case)
Peak current consumption	<30 mA	<15 mA
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, gaming, headsets, stereo audio streaming, automotive, PCs, security, proximity, healthcare, sports & fitness, etc.	Mobile phones, gaming, PCs, watches, sports and fitness, healthcare, security & proximity, automotive, home electronics, automation, Industrial, etc.

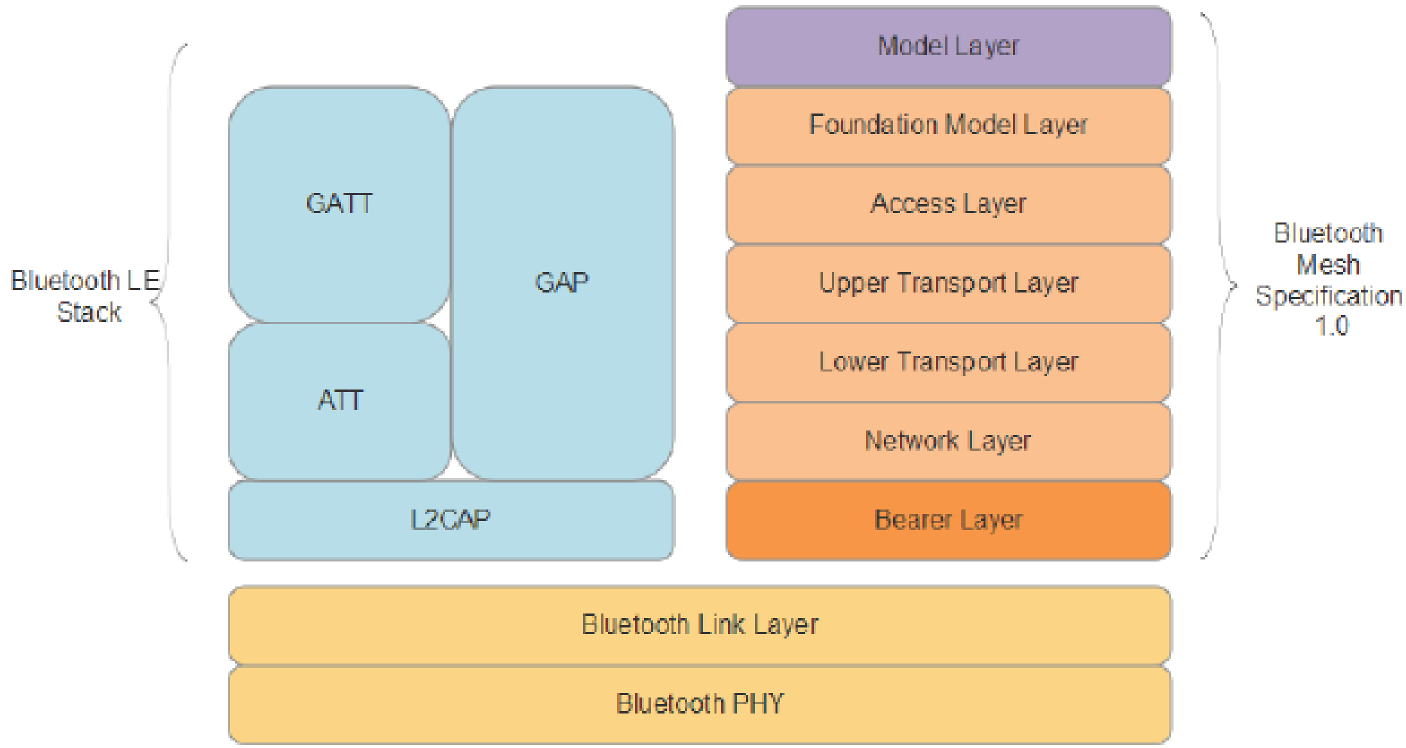
Bluetooth & BLE

- Bluetooth & BLE topologies



BLE & Bluetooth Mesh Spec

- Bluetooth Mesh Specification 1.0 Stack



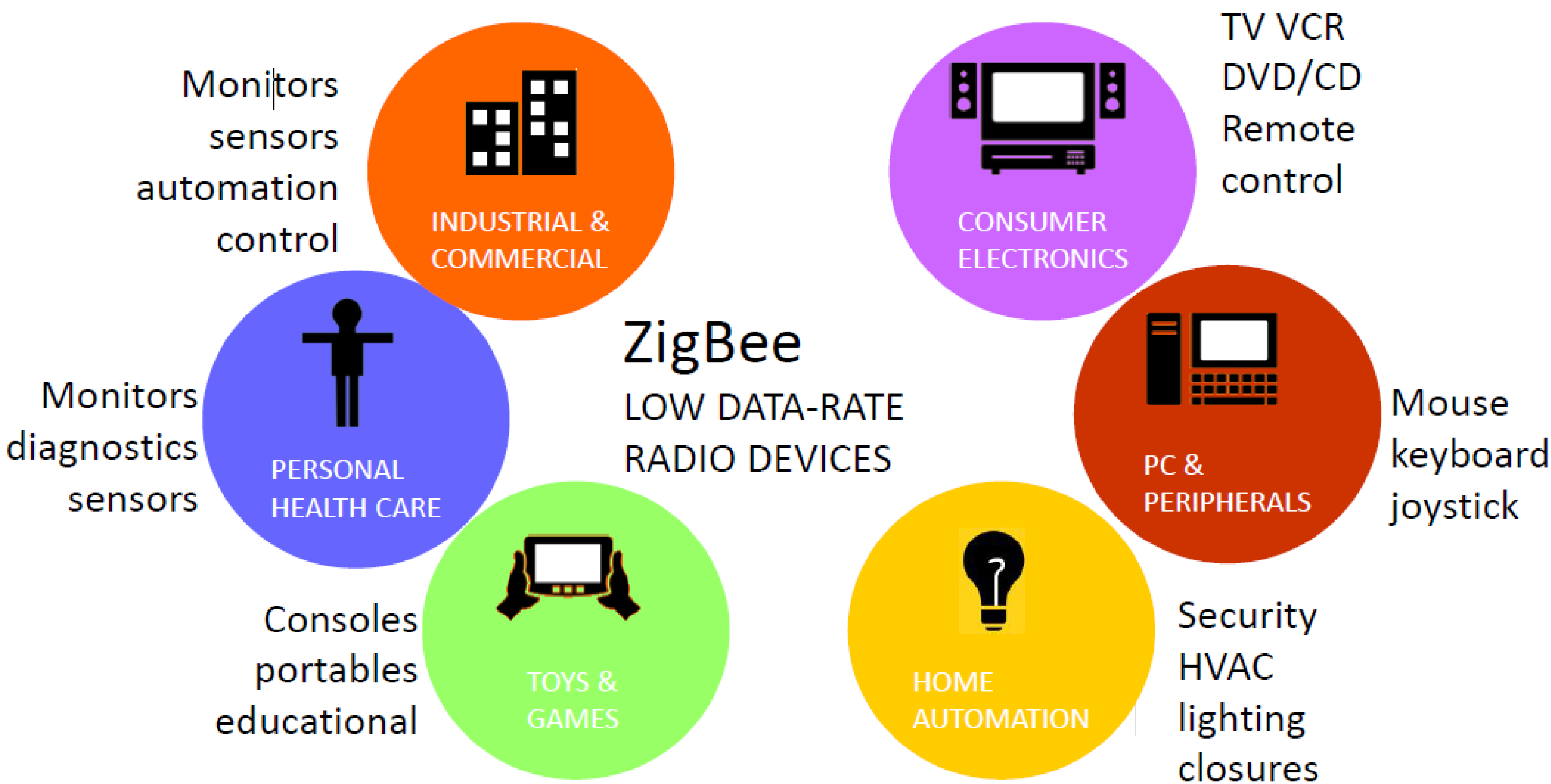
ZigBee - Introduction

- ZigBee is a technological standard designed for control and sensor networks
- Based on the IEEE 802.15.4 Standard
- Created by the ZigBee Alliance
- Operates in Personal Area Networks (PAN's) and device-to-device networks
- Connectivity between small packet devices
- Control of lights, switches, thermostats, appliances, etc.

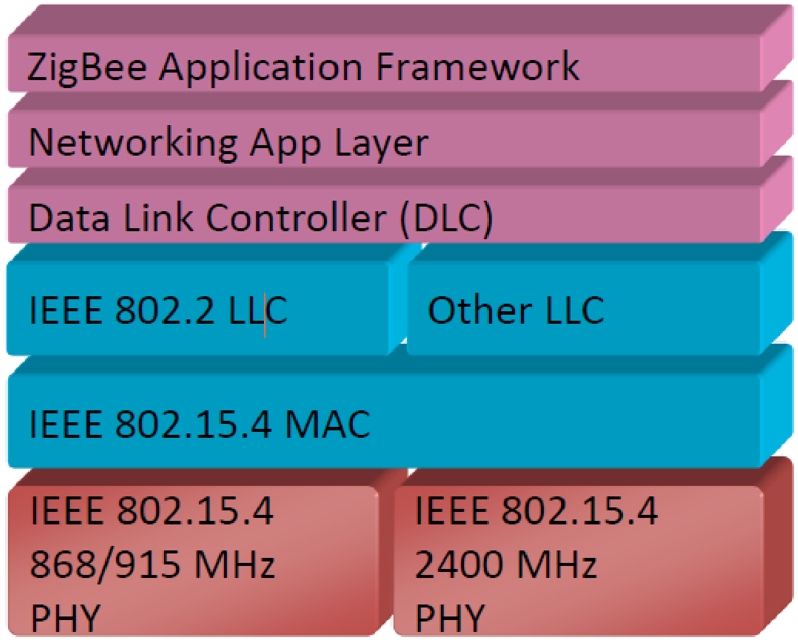
ZigBee - Characteristics

- Low cost
- Low power consumption
- Low data rate
- Relatively short transmission range
- Scalability
- Reliability
- Flexible protocol design suitable for many applications

ZigBee - Applications



ZigBee/IEEE 802.15.4 Architecture and Frame Format



Sync Header		PHY Header		PHY Payload
Preamble	Start of Packet Delimiter	Frame Length (7 bit)	Reserve (1 bit)	PHY Service Data Unit (PSDU)
4 Octets	1 Octets	1 Octets		0-127 Bytes

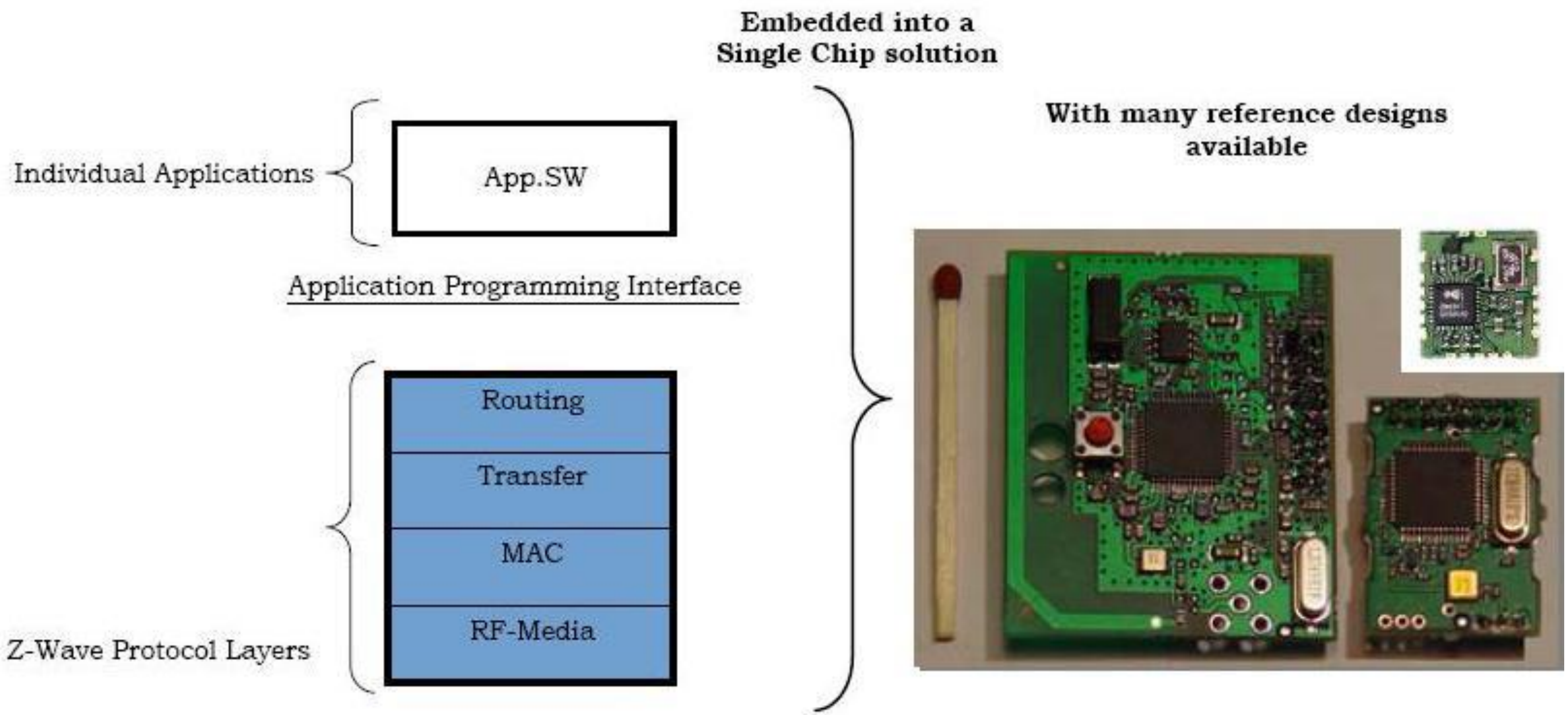
ZigBee and other Protocols

Feature(s)	IEEE 802.11b	Bluetooth	ZigBee
Power Profile	Hours	Days	Years
Complexity	Very Complex	Complex	Simple
Nodes/Master	32	7	64000
Latency	Enumeration up to 3 Seconds	Enumeration up to 10 seconds	Enumeration 30ms
Range	100 m	10m	70m-300m
Extendibility	Roaming Possible	No	YES
Data Rate	11Mbps	1 Mbps	250Kbps
Security	Authentication Service Set ID (SSID), WEP	64 bit, 128 bit	128 bit AES and Application Layer user defined

Z-Wave - Introduction

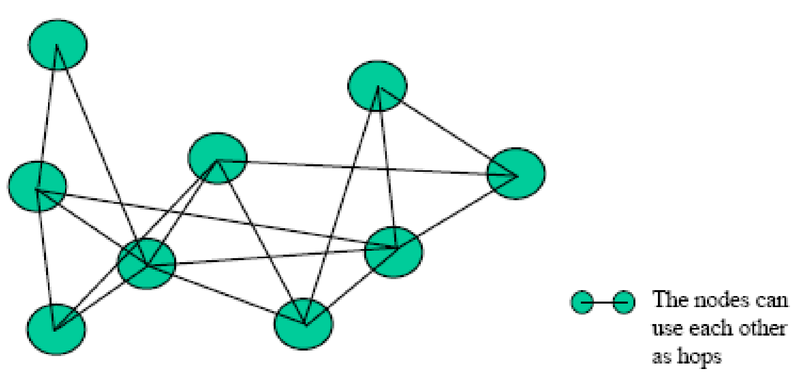
- Zensysa Danish-American company founded in 1999 invented the Z-wave technology.
- They are basically providers of Integrated Single chip Solutions.
- While trying to embed intelligence and RF communication into their products they stumbled upon the idea to come up with a new technology combining the pros of the existing technologies.

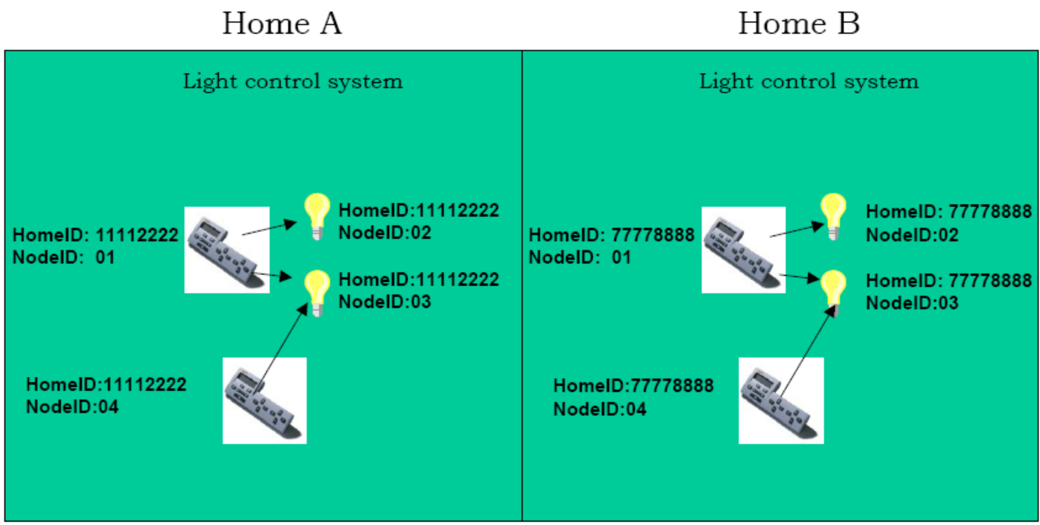
Z-Wave - Protocol layers



Z-Wave - Characteristics

- The Z-Wave Network is of the mesh architecture.
- Efficiency of the Z-Wave Network is because of the Routing Protocol it uses.
- More than one Z-Wave Network can co-exist.
- A Z-Wave network can consist of 232 nodes to the max.
- Typical Z-Wave Network





Z-Wave vs. ZigBee

- Comparison between Z-Wave and ZigBee

	Properties	Z-Wave	ZigBee
Physical Layer	RF band (MHz)	868/908 (all chips) 2400 (400 series chip)	868/915/2400
	Range (m)	30 (indoors) 100 (outdoors)	10-100
	Bit rate (kb/s)	9.6/40(200 series chip) 200 (only 400 series chip)	20/40/250
	Receiver sensitivity (dBm)	-101 (at 40kb/s)	-85 (2.4GHz band) -92 (868/915 MHz bands)

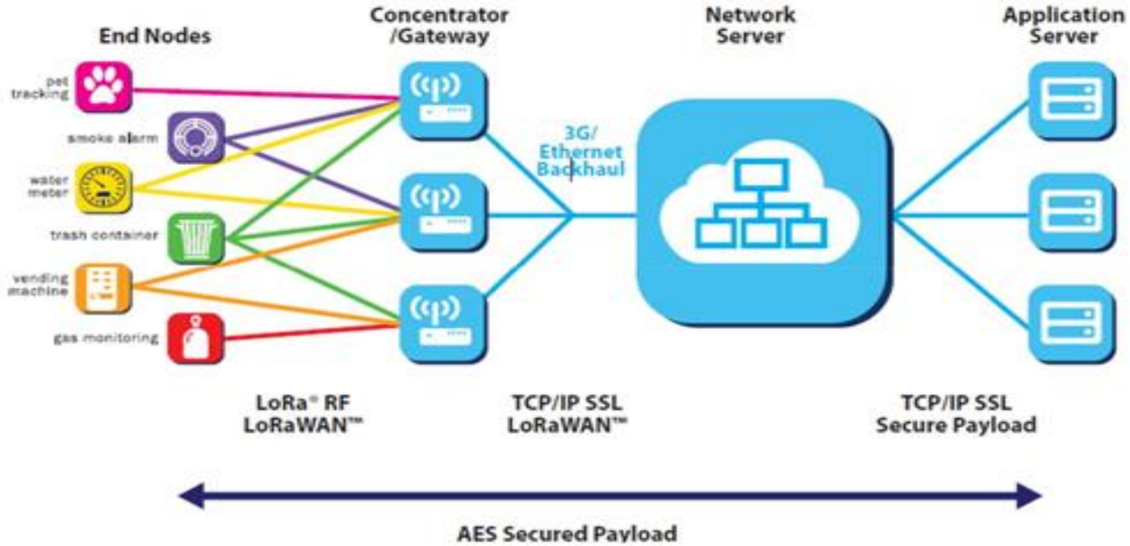
LoRaWAN - Introduction

- An open standard architecture developed by LoRa Alliance
- Provide a medium access control mechanism and enable End-Devices to communicate with one or more gateways
- Physical layer technology that enables long range, low data rate, and low power wireless communication
- LoRaWAN constitutes a data link layer protocol above the LoRa physical layer protocol

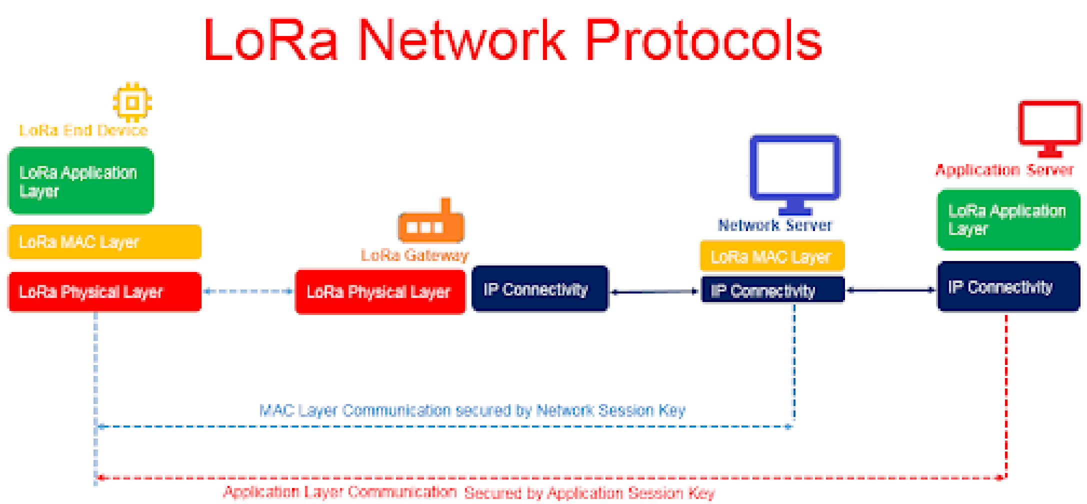
Specification	**LoRa Technology Support**
Standard	LoRa Alliance
Operational Frequencies	Unlicensed ISM band 868, 915 MHz
Modulation	Chirp spread spectrum (CSS)
Coverage Range (Km)	2 - 5 (urban) / 15 (rural)
Data Rate (kbps)	0.3 - 50 (EU) / 0.9 - 100 (US)
Topology	Star

LoRaWAN - Architecture and Communication

- Architecture



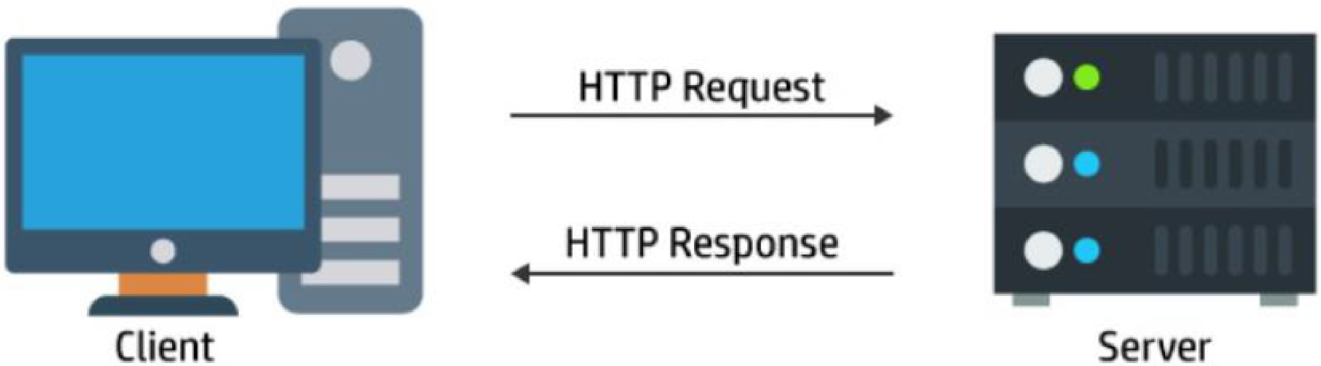
- Communication

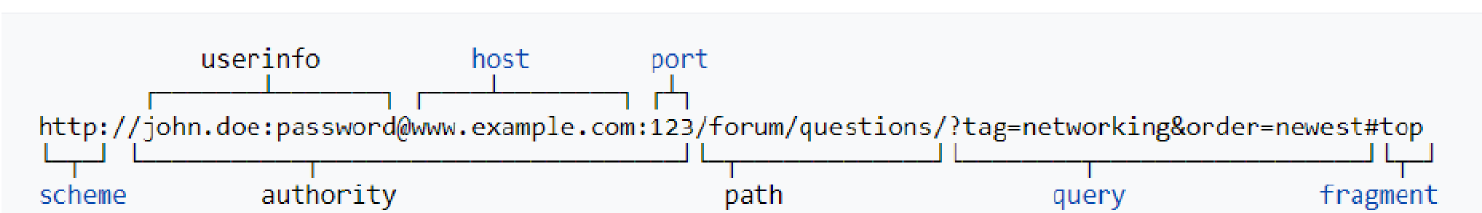


APPLICATION LAYER PROTOCOLS

HTTP

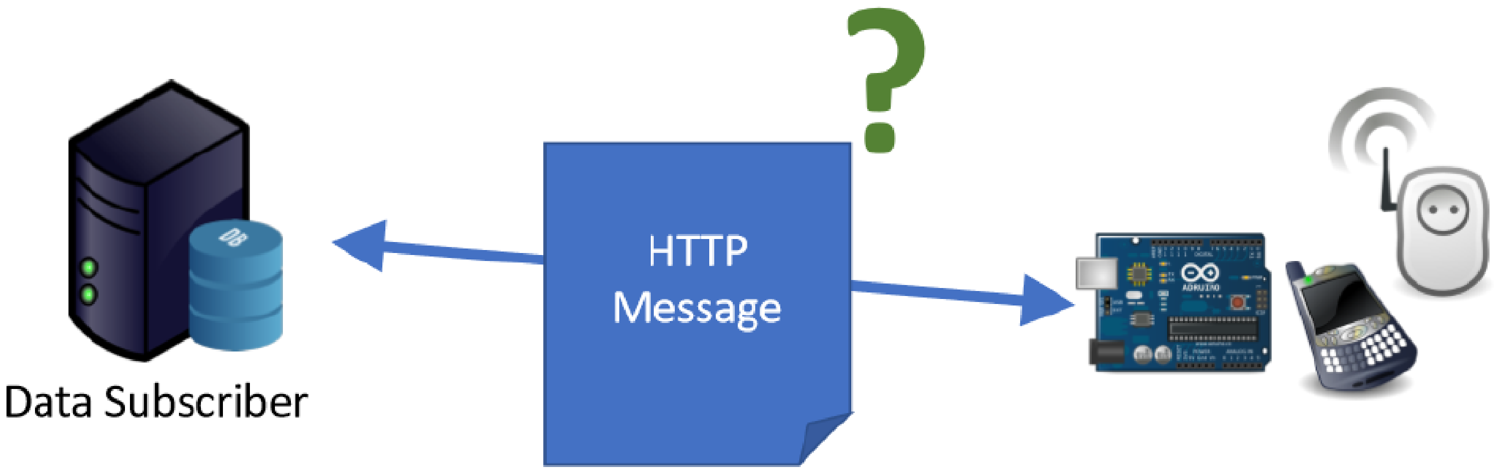
- HTTP is connectionless
- HTTP is Simple
- HTTP is extensible
- HTTP is stateless but not sessionless





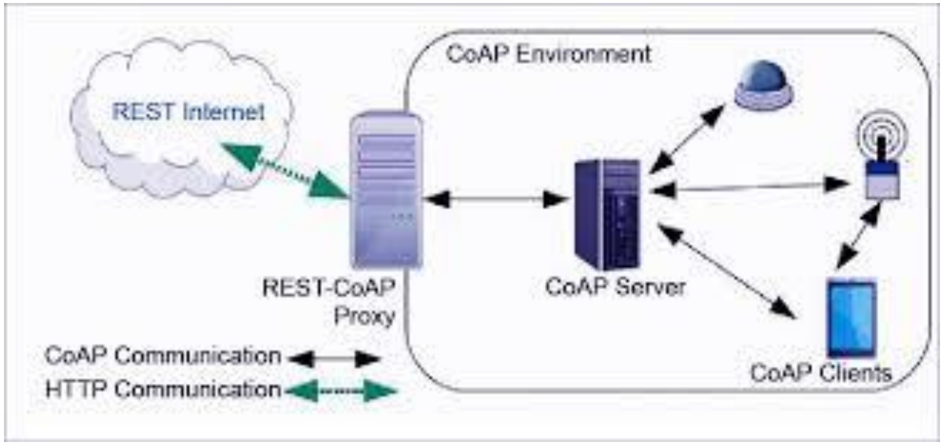
Why not HTTP?

- **Slower:** because it uses bigger data packets to communicate with the server
- **Overhead:** HTTP request opens and closes the connection at each request
- **Power consuming:** since it takes a longer time and more data packets, therefore it uses much power

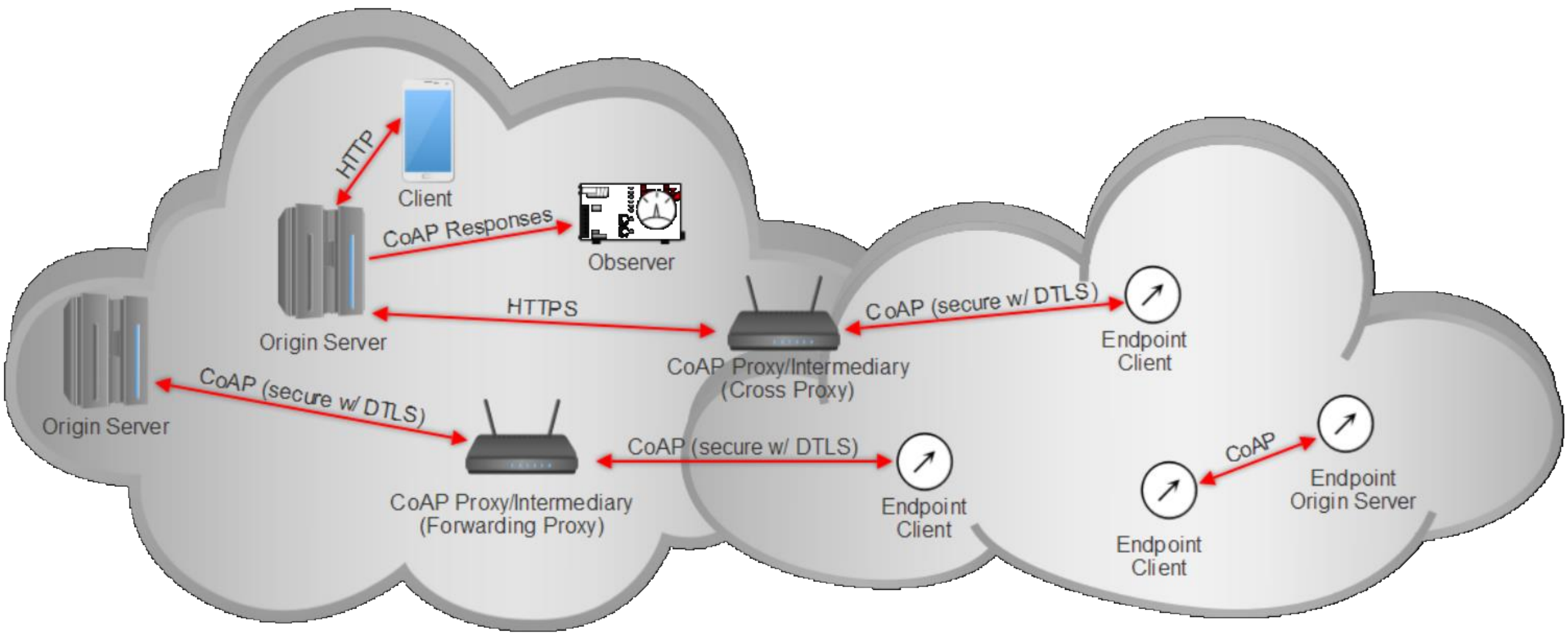


COAP

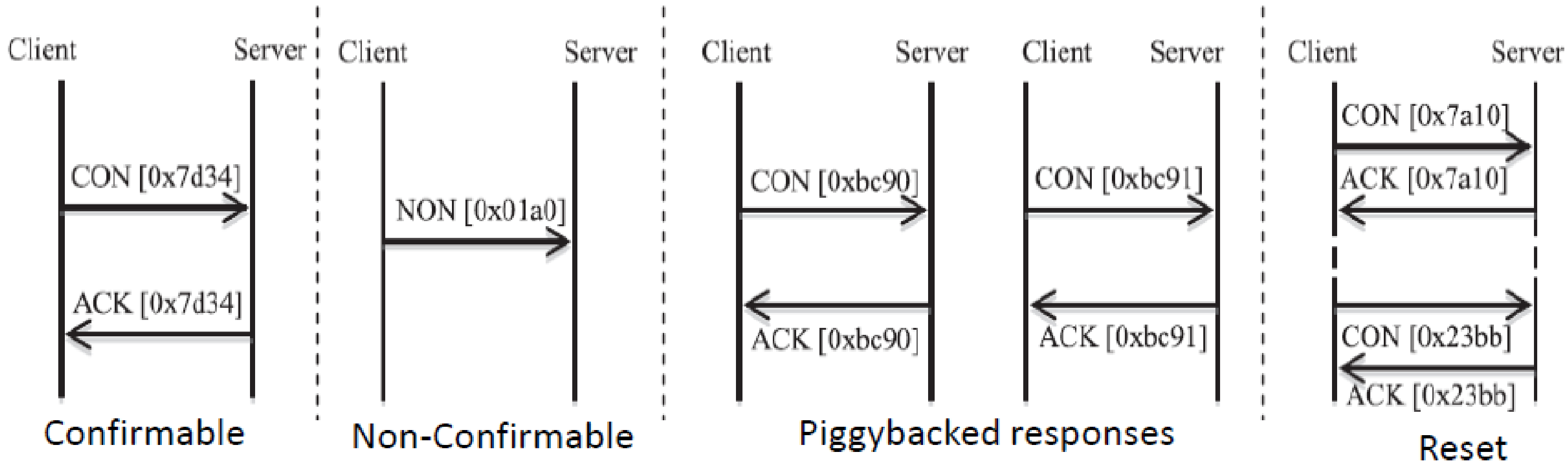
- Constrained Application Protocol - IETF (RFC7228)
- Defines a web transfer protocol based on *Representational State Transfer* (REST) on top of HTTP functionalities
- Used for M2M applications
- Follow REQUEST/RESPONSE model
- Runs on top of UDP
- Command: GET,PUT,POST,DELETE



COAP Architecture



CoAP - Message Format & Type



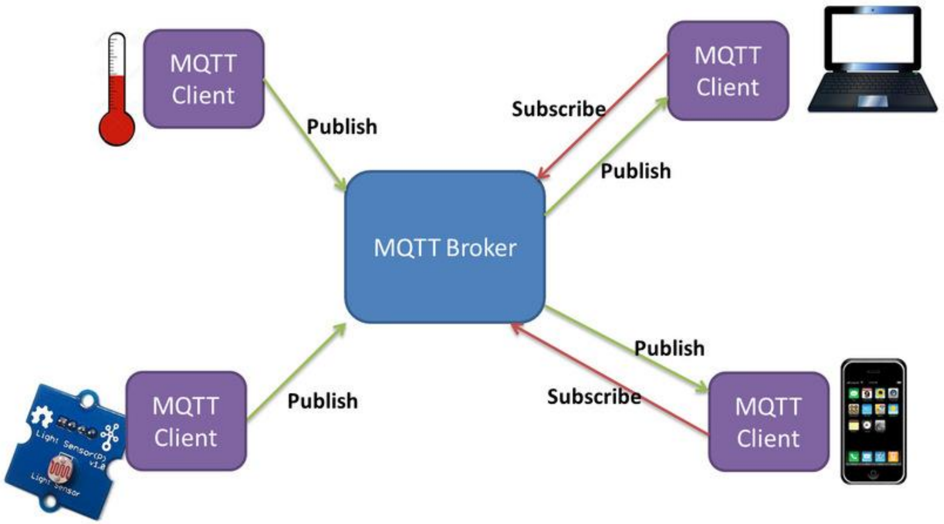
0 1	2 3	4 5 6 7	8	16 31
Ver	T	OC	Code	Message ID
Token (if any)				
Options (if any)				
Payload (if any)				

COAP - Features

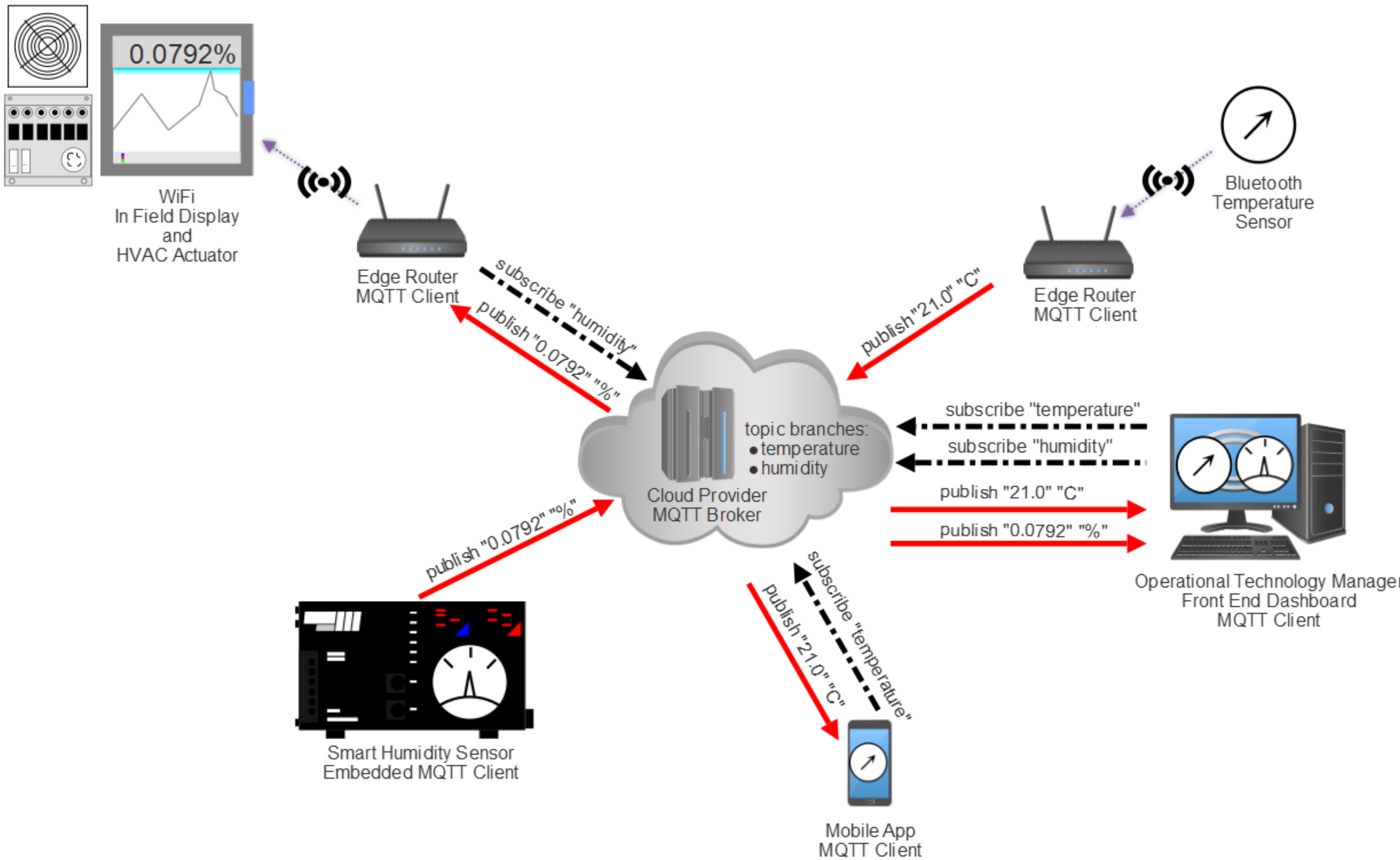
- ***Resource observation*:** On-demand subscriptions to monitor resources of interest using publish/subscribe mechanism.
- ***Block-wise resource transport*:** Ability to exchange transceiver data between the client and the server without the need to update the whole data to reduce the communication overhead.
- ***Resource discovery*:** Server utilizes well-known URI paths based on the web link fields in CoRE link format to provide resource discovery for the client.
- ***Interacting with HTTP*:** Flexibility of communicating with several devices because the common REST architecture enables CoAPto interact easily with HTTP through a proxy.
- ***Security*:** CoAP is a secure protocol since it is built on top of *datagram transport layer security* to guarantee integrity and confidentiality of exchanged messages.

MQTT

- *Message Queue Telemetry Transport*
- Messaging protocol introduced by IBM
- MQTT aims at connecting embedded devices and networks with applications and middleware
- Used for M2M applications
- Publish/Subscribe model
- Runs on top of TCP
- MQTT simply consists of three components, subscriber, publisher, and broker

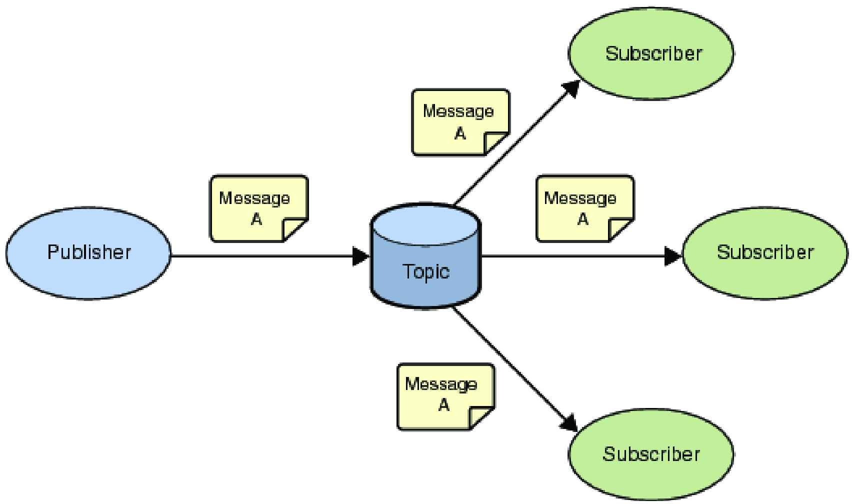


MQTT - Architecture



MQTT - Publish Subscribe Messaging

- A Publish Subscribe messaging protocol allowing a message to be published once and multiple consumers (applications / devices) to receive the message providing decoupling between the producer and consumer(s)
- A producer sends (publishes) a message (publication) on a topic (subject)
- A consumer subscribes (makes a subscription) for messages on a topic (subject)
- A message server / broker matches publications to subscriptions
  - If no matches the message is discarded
  - If one or more matches the message is delivered to each matching subscriber/consumer



MQTT - Publish Subscribe Messaging

- A topic forms the namespace
- Is hierarchical with each "subtopic" separated by a /
- An example topic space
  - A house publishes information about itself on:

    <country>/<region>/<town>/<postcode>/<house>/energyConsumption
    <country>/<region>/<town>/<postcode>/<house>/solarEnergy
    <country>/<region>/<town>/<postcode>/<house>/alarmState
    <country>/<region>/<town>/<postcode>/<house>/alarmState
  - *And subscribes for control commands:*

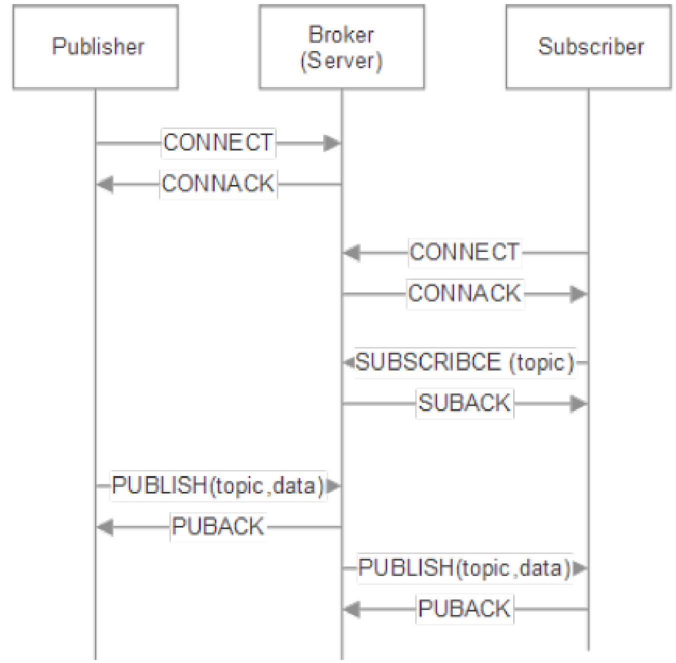
    <country>/<region>/<town>/<postcode>/<house>/thermostat/setTemp
- A subscriber can subscribe to an absolute topic or can use wildcards:
  - Single-level wildcards "+" can appear anywhere in the topic string
  - Multi-level wildcards "#" must appear at the end of the string
  - Wildcards must be next to a separator
  - Cannot be used wildcards when publishing

MQTT - Publish Subscribe Messaging

- A subscription can be durable or non-durable
  - Durable:
    - Once a subscription is in place a broker will forward matching messages to the subscriber:
    - Immediately if the subscriber is connected
    - If the subscriber is not connected messages are stored on the server/broker until the next time the subscriber connects
  - Non-durable: The subscription lifetime is the same as the time the subscriber is connected to the server / broker
- A publication may be retained
  - A publisher can mark a publication as retained
  - The broker / server remembers the last known good message of a retained topic
  - The broker / server gives the last known good message to new subscribers
    - i.e. the new subscriber does not have to wait for a publisher to publish a message in order to receive its first message

MQTT - Communication

- Communication Mode

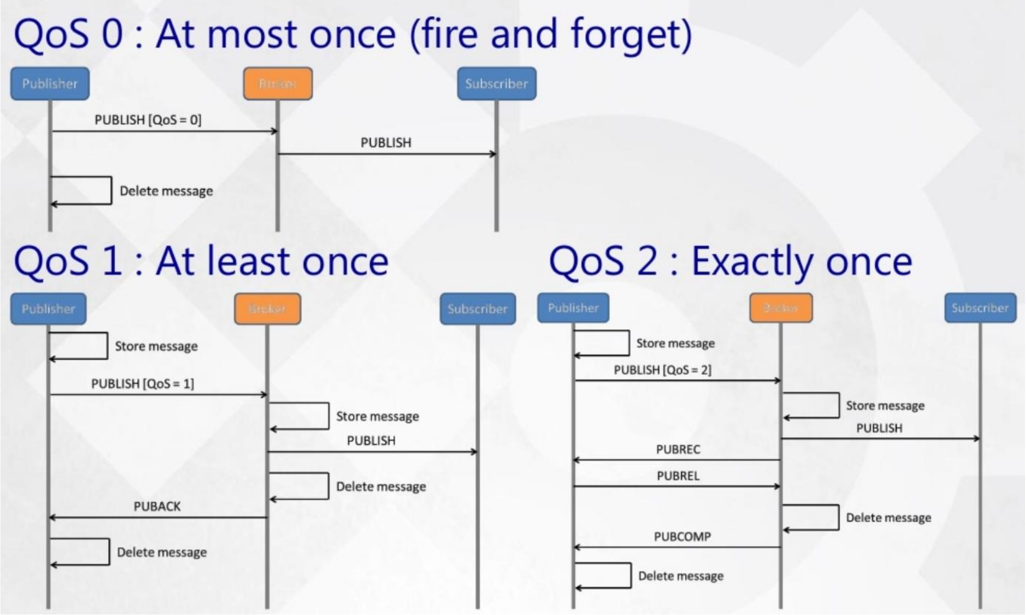


- Message Format

0	1	2	3	4	5	6	7
Message Type				UDP	QoS Level		Retain
Remaining Length (1~4 bytes)							
Variable Length Header (Optional)							
Variable Length Message Payload (Optional)							

MQTT - QoS Level

- **At most once** -the message is sent only once and the client and broker take no additional steps to acknowledge delivery
- **At least once** -the message is re-tried by the sender multiple times until acknowledgement is received
- **Exactly once** -the sender and receiver engage in a two-level handshake to ensure only one copy of the message is received

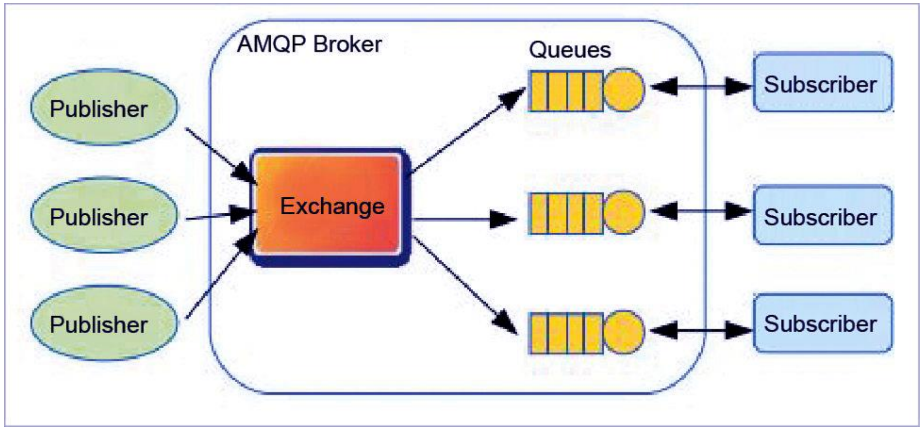


MQTT - Advantages

- **Open Standard:** Open specification and standard, 40+ clients implementation
- **Lightweight:** Minimum overhead, Efficient Format, Tiny client
- **Reliable:** QoS for reliability on unreliable network
- **Simple:** Simple documentation, support subscriber, publisher, and broker
- **A lot of implementations:** open source, cloud and commercial

AMQP

- *Advanced Message Queuing Protocol*
- An open standard application layer protocol for focusing on message-oriented environments
- Communications are handled by two main components- exchanges and message queues
- Exchanges are used to route the messages to appropriate queues.
- Messages can be stored in message queues and then be sent to receivers

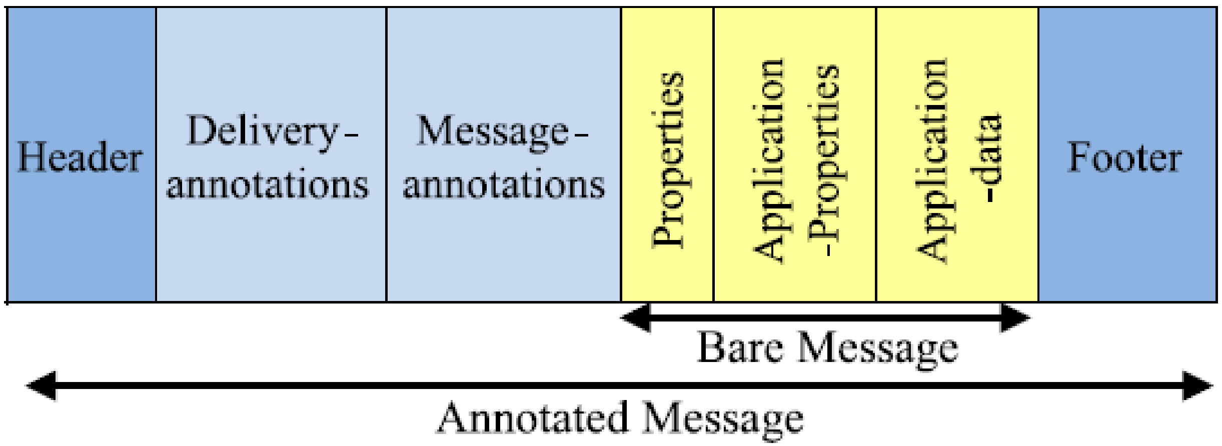


AMQP - Overlay Network

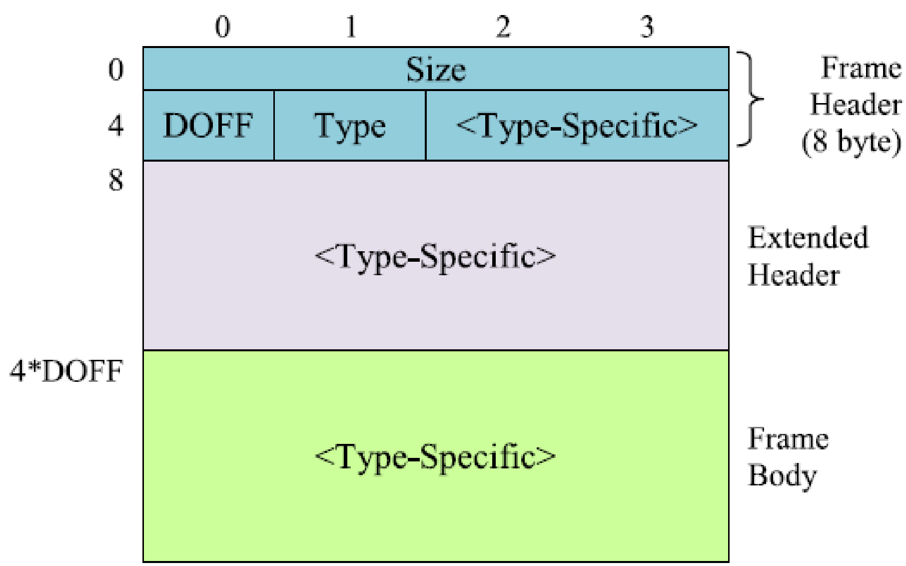
- **Broker**
  - Applications Connect to a Broker to participate in the AMQP network
  - The Connection is used to establish a Session
  - Sessions provide state between Connections, establish identity, ease failover
  - Connections are further subdivided into Channels
  - Multiple threads of control within an Application can share one Connection
- **Queues**
  - Applications logic interacts ONLY with Queues
  - Queues have well known Names - Addressable
  - Applications do not need to know how messages get in/out of Queues
  - Queues can be smart, they are an extension point
  - Applications will assign implied semantics to Queues (e.g. “StockOrderQueue”)
- **Links**
  - Links move Messages between Queues and/or Applications
  - Contain Routing and Predicate Evaluation Logic –similar to Complex Event Processing

AMQP Communication

- Message



- Frame format



AMQP - Advantages

- **Lightweight**
  - Minimum overhead, Efficient Format, Tiny client
- **Reliable**
  - QoS for reliability on unreliable network
- **Secure**
  - Provide reliable communication link
- **Open Standard**
  - Open specification and standard
- **Interoperability**
  - Support heterogeneous set of computing devices

IoT Application layer protocols

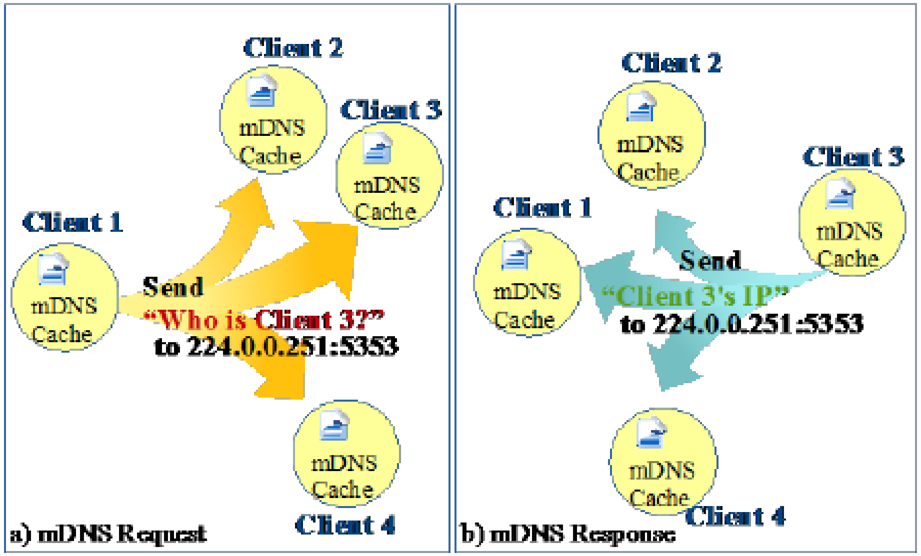
- Summary & comparison

	MQTT	MQTT-SN	CoAP	AMQP	STOMP	HTTP/RESTful
Model	MOM pub/sub	MOM pub/sub	RESTful	MOM	MOM	RESTful
Discovery protocol	No	Yes (via gateways)	Yes	No	No	Yes
Resource demands	Low	Very Low	Very Low	High	Medium	Very High
Header Size (bytes)	2	2	4	8	8	8
Average power usage	Lowest	Low	Medium	High	Medium	High
Authentication	No (SSL/TLS)	No (/TLS)	No (DTLS)	Yes	No	Yes (TLS)
Encryption	No (SSL/TLS)	No (SSL/TLS)	No (DTLS)	Yes	No	Yes (TLS)
Access controls	No	No	No (proxy)	Yes	No	Yes
Communication overhead	Low	Very Low	Very Low	High	High, verbose	High
Protocol complexity	Low	Low	Low	High	Low	Very High
TCP/UDP	TCP	TCP/UDP	UDP	TCP/UDP	TCP	TCP
Broadcasting	Indirect	Indirect	Yes	No	No	No
Quality of Service	Yes	Yes	With CON messages	Yes	No	No

SERVICE DISCOVERY PROTOCOLS

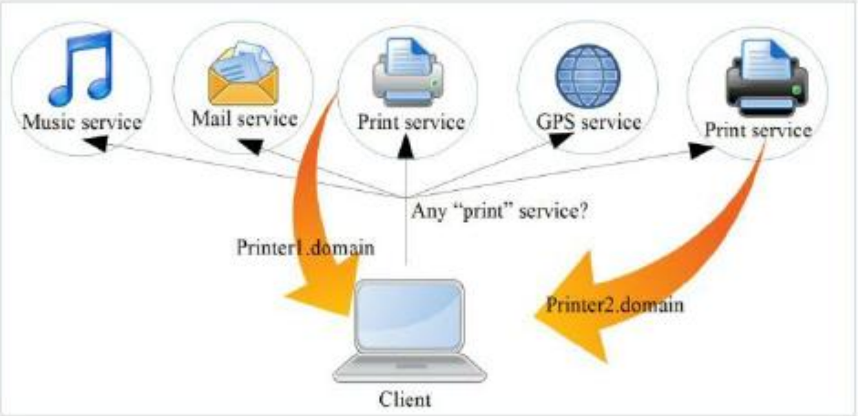
mDNS - Multicast DNS

- Flexible protocol
- There is no need for manual reconfiguration or extra administration to manage devices
- It can run without infrastructure
- It can continue working if failure of infrastructure happens
- mDNS inquires names by sending an IP multicast message to all the nodes in the local domain



DNS-SD - DNS Service Discovery

- The pairing function of required services by clients using mDNS
- Clients can discover a set of desired services in a specific network by employing standard DNS messages
- Zero configuration aids to connect machines without external administration or configuration
- There are two main steps to process Service Discovery
  1. Finding host names of required services
  2. pairing IP addresses with their host names using mDNS



References

- Perry Lea, ***IoT and Edge Computing for Architects,*** 2nd Edition, Packt Publishing, 2020
  - Chapter 5: Non-IP Based WPAN
  - Chapter 6: IP-Based WPAN and WLAN
  - Chapter 7: Long Range Communication Systems and Protocols (WAN)
  - Chapter 10: Edge to Cloud Protocols
- A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*, IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.