Advanced Internet of Things Technologies

Designed by: Thuat NGUYEN-KHANH

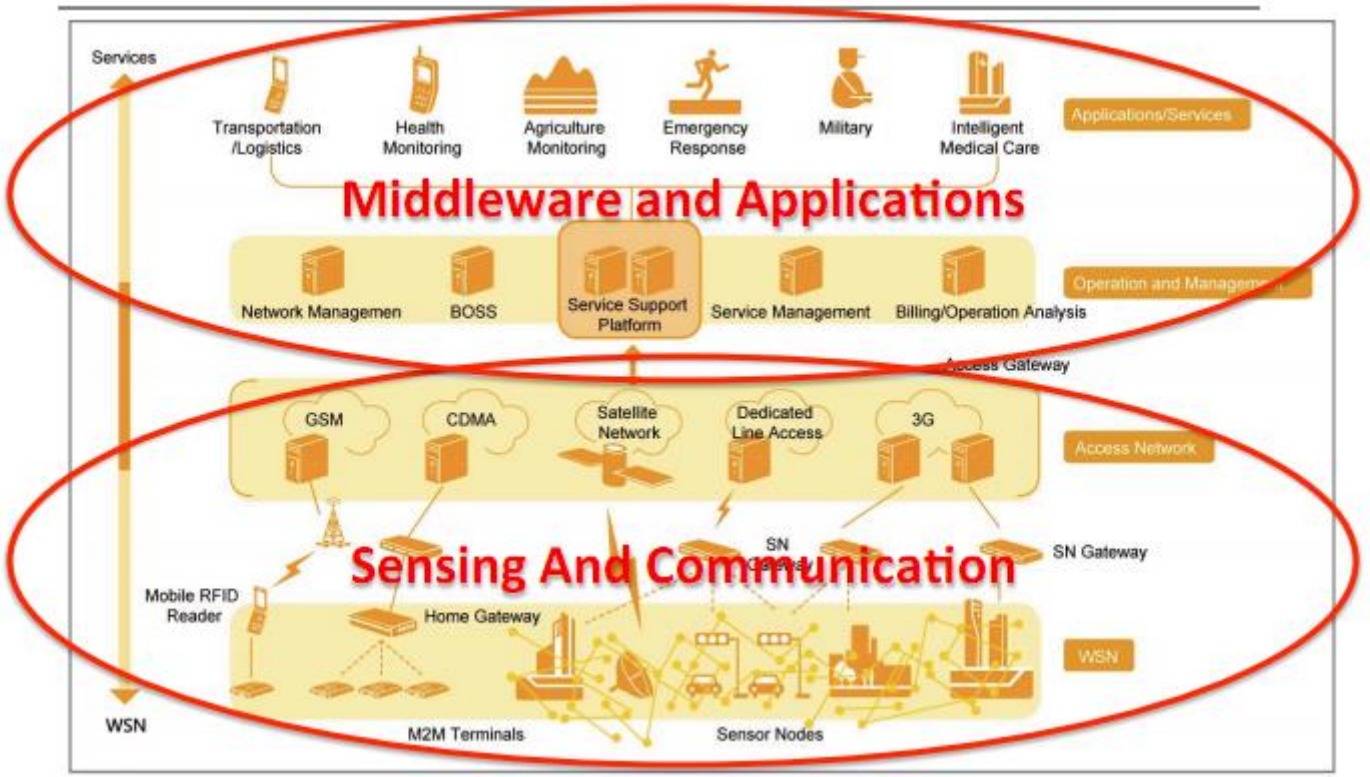
Lecturer at The Faculty of Computer Networks & Communications - UIT - VNU-HCM

Email: thuatnk@uit.edu.vn

Chapter 6: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

IoT Layered Architecture



- Source: ZTE

Chapter 5: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

Problem definition

- Distributed network of devices communicating with a central location or to each other.
- Devices that run on batteries or with limited power.
- Information flows over unreliable networks (cellular, satellite, any wireless technology in general).
- No need to write an application protocol from scratch on top of TCP/IP.

Chapter 5: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

IoT Protocols

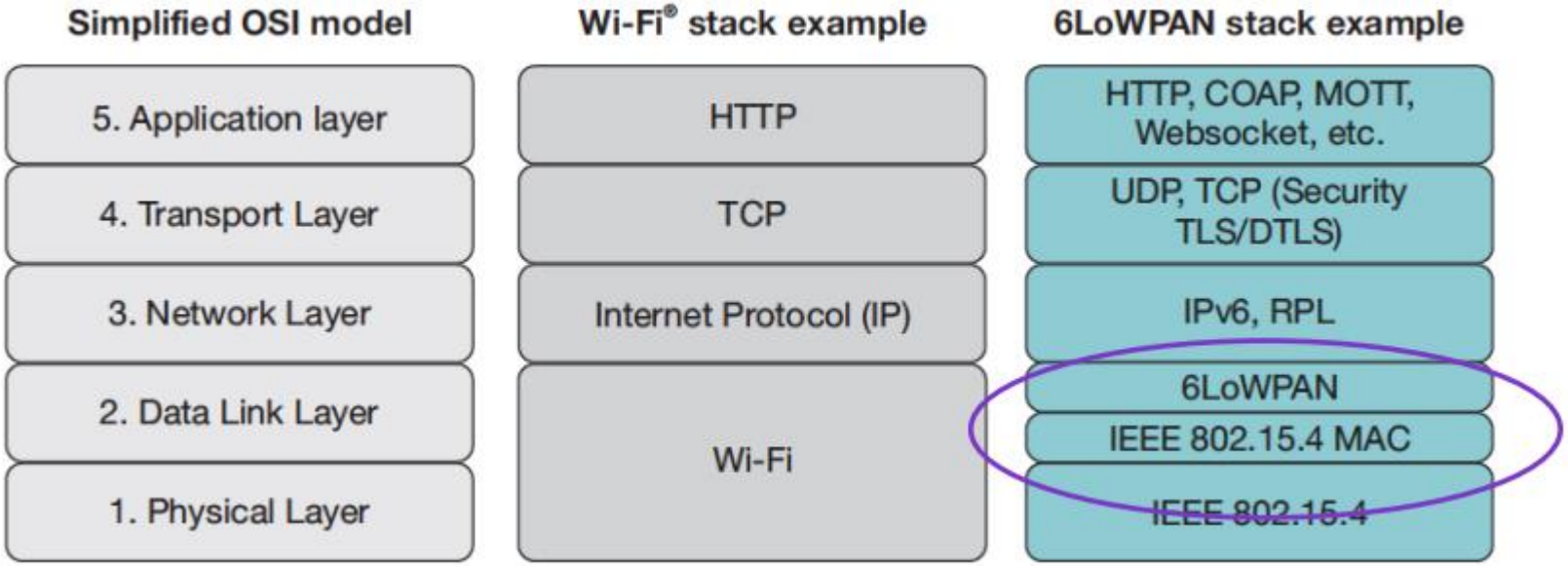
Session		**MQTT**, **SMQTT**, **CoRE**, **DDS**, **AMQP** , **XMPP**, **CoAP**, …	**Security**	**Management**
Network	**Encapsulation**	**6LowPAN**, **6TiSCH**, **6Lo**, Thread, …	TCG, Oath 2.0, SMACK, SASL, ISASecure, ace, DTLS, Dice, …	**IEEE 1905**, **IEEE 1451**, …
	Routing	**RPL**, **CORPL**, **CARP**, …		
Datalink		WiFi, **Bluetooth Low Energy**, **Z-Wave**, **ZigBee Smart**, **DECT/ULE**, 3G/LTE, NFC, **Weightless**, **HomePlug GP**, **802.11ah**, **802.15.4e**, **G.9959**, **WirelessHART**, **DASH7**, ANT+, **LTE-A**, **LoRaWAN**, …		

Chapter 5: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

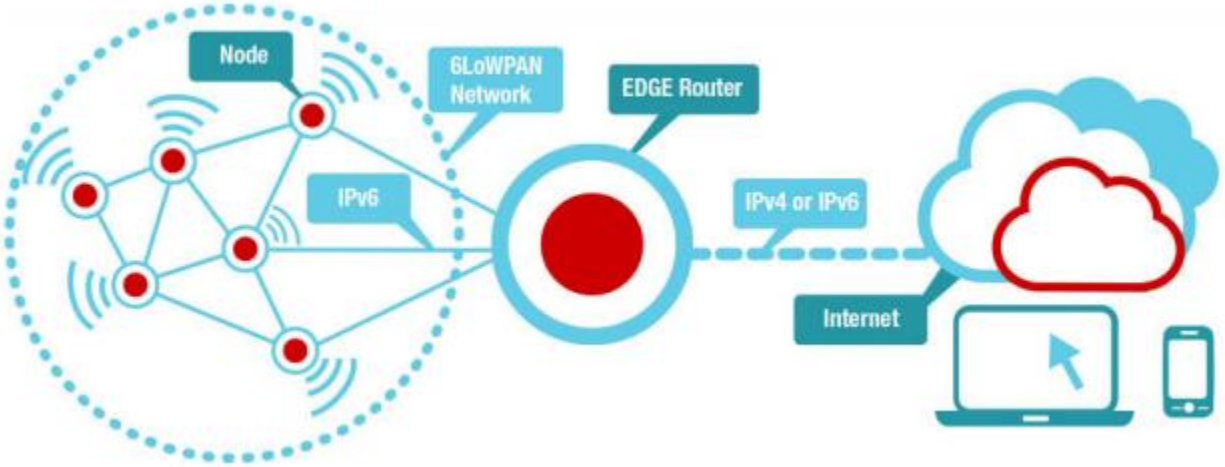
6LoWPAN

- 6LoWPAN: IPv6 over Low Power Wireless Personal Area Network



6LoWPAN

- TCP/IP is not one size fits all
- 6LoWPAN: Try to optimize the transmission of IPv6 packets over low power and lossy network such as 802.15.4



6LoWPAN

- RFC6282:
  - **Header compression:** compresses the 40-byte IPv6 and 8-byte UDP headers by assuming the usage of common fields
  - **Fragmentation and reassembly:** The data link of IEEE 802.15.4 with a frame length of maximum 127 bytes does not match the MTU of IPv6, which is 1280 bytes.
  - **Stateless auto configuration:** the process where devices inside the 6LoWPAN network automatically generate their own IPv6 address.

Chapter 5: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

Introduction to MQTT (1/2)

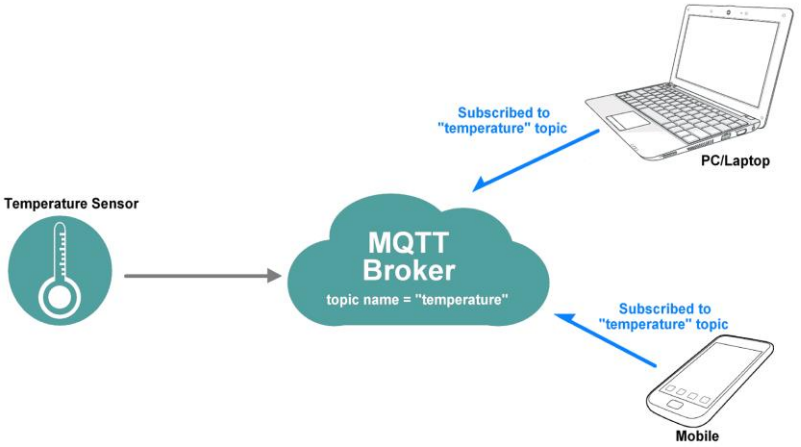
- Message Queuing Telemetry Transport
- MQTT is a lightweight publish/subscribe protocol with reliable bi-directional message delivery.
- Invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper. The original problem was to send sensor data from oil pipelines through a satellite link.

Introduction to MQTT (2/2)

- Invented and sponsored by IBM. Now Open source. Open Source libraries available.
- Lightweight = Low network bandwidth and small code footprint.
- Expect that client applications may have very limited resources available (8 bit controller, 256kb ram)

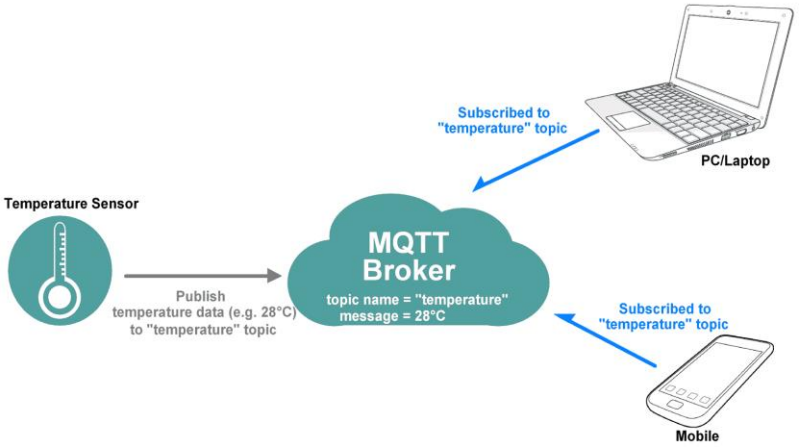
Publish/Subscribe Messaging Model (1/3)

- Clients subscribe to topics(SUBSCRIBE)



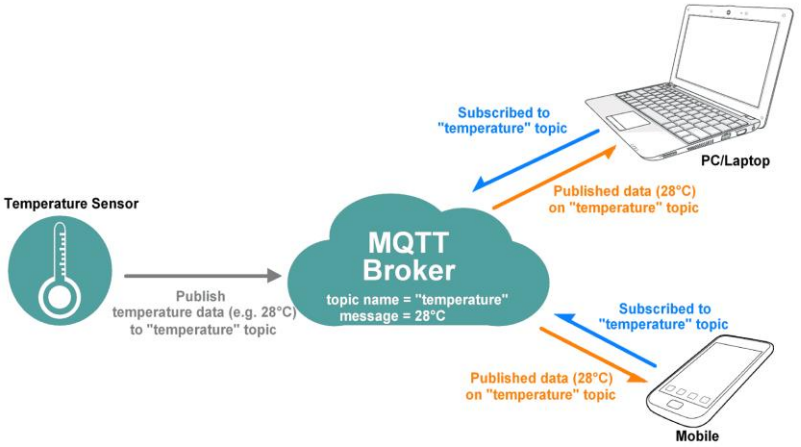
Publish/Subscribe Messaging Model (2/3)

- Clients subscribe to topics(SUBSCRIBE)
- Messages are published to a specific topic name (PUBLISH)



Publish/Subscribe Messaging Model (3/3)

- Clients subscribe to topics(SUBSCRIBE)
- Messages are published to a specific topic name (PUBLISH)
- A broker server handles the routing of messages



MQTT – QoS (1/4)

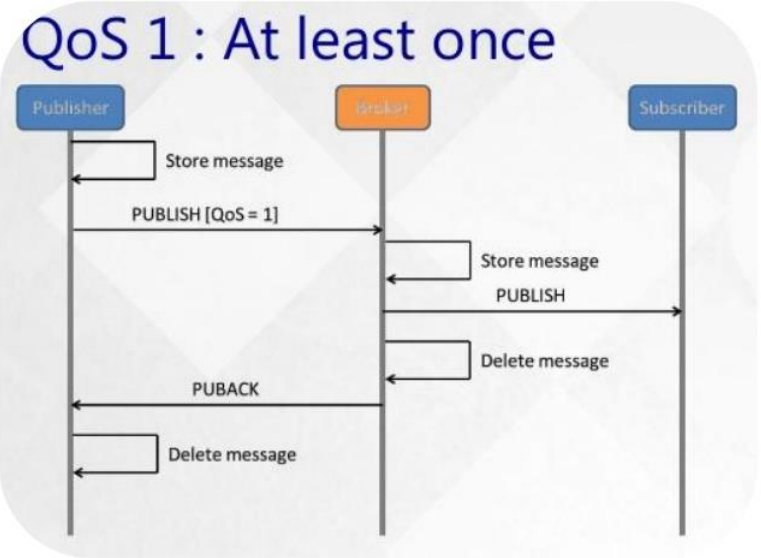
- The Quality of Service used to deliver a message
  - 0: Best effort
    - PUBLISH
  - 1: At least once
    - PUBLISH + PUBACK
  - 2: Exactly once
    - PUBLISH + PUBREC + PUBREL + PUBCOMP

MQTT – QoS (2/4)

- The Quality of Service used to deliver a message
  - 0: Best effort
    - PUBLISH

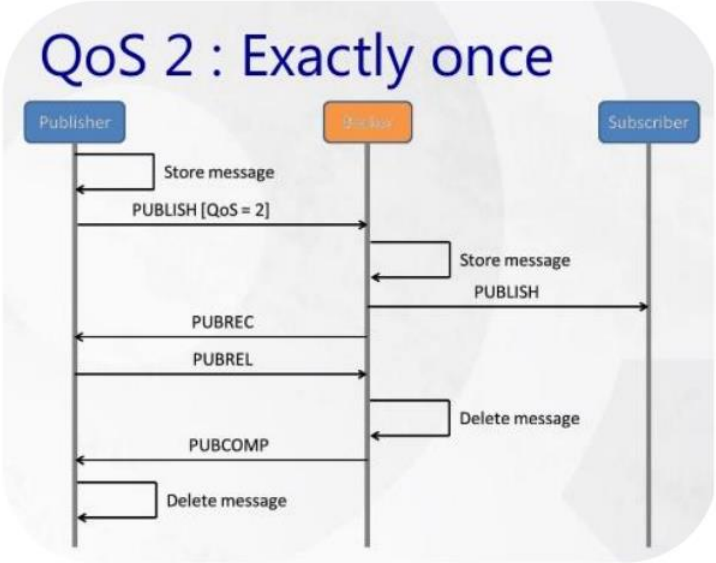
MQTT – QoS (3/4)

- The Quality of Service used to deliver a message
  - 1: At least once
    - PUBLISH + PUBACK



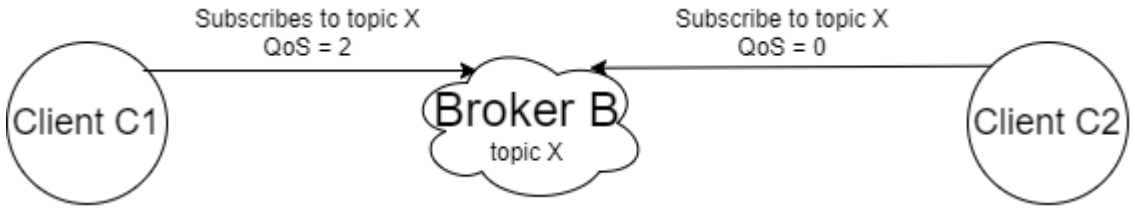
MQTT – QoS (4/4)

- The Quality of Service used to deliver a message
  - 2: Exactly once
    - PUBLISH + PUBREC + PUBREL + PUBCOMP

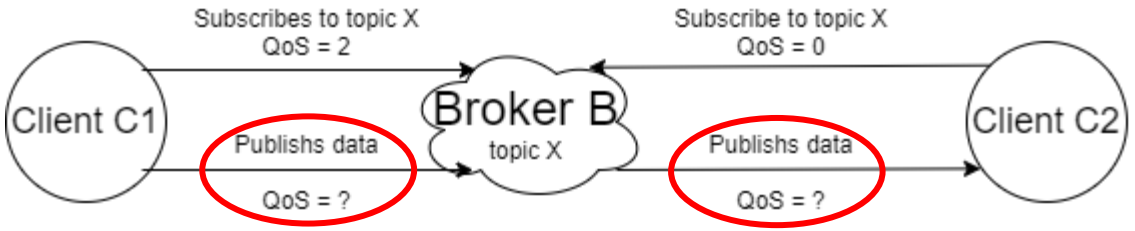


MQTT – QoS Example 1

- If Client C1 subscribes to Broker B with QoS = 2; Client C2 QoS = 0

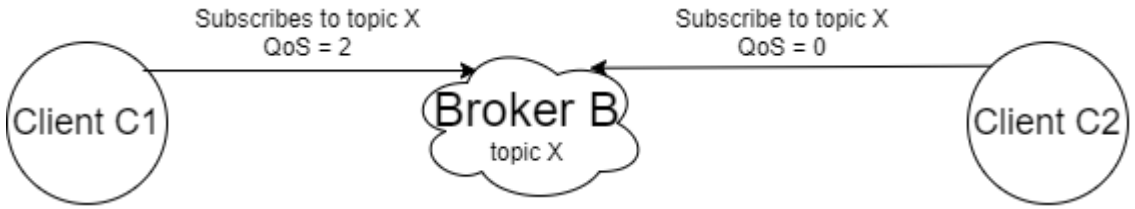


- When C1 publishs data to Broker.
- What is QoS if Broker publishs data to C2?

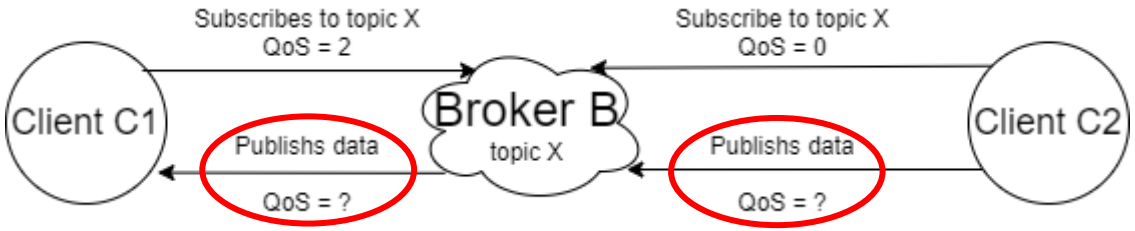


MQTT – QoS Example 2

- If Client C1 subscribes to Broker B with QoS = 2; Client C2 QoS = 0



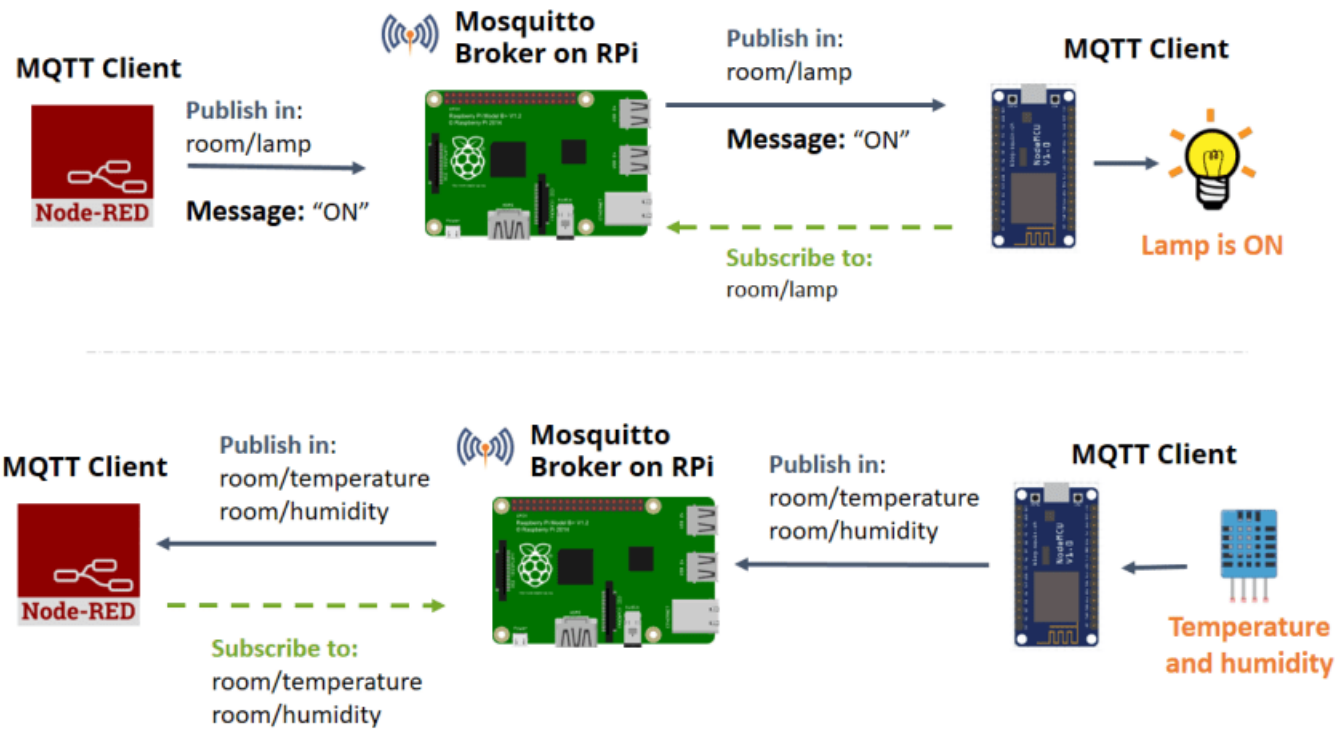
- When C2 publishs data to Broker.
- What is QoS if Broker publishs data to C1?



MQTT vs HTTP

	MQTT	HTTP
Design	Data centric	Document centric
Pattern	Publish/Subscribe	Request /Response
Complexity	Simple	More Complex
Message Size	Small. Binary with 2B header	Large. ASCII
Service Levels	Three	One
Libraries	30kB C and 100 kB Java	Large
Data Distribution	1 to zero, one, or n	1 to 1 only

MQTT – A case study



- Source: https://randomnerdtutorials.com/esp8266-and-node-red-with-mqtt/

Some MQTT Client

- **MQTT CLI** is an open-source Java MQTT client tool (Link)
- **HiveMQ MQTT Client** is a Java library that is available under the Apache license on GitHub (Link)
- **MQTT.fx** is implemented with JavaFX (available for Win/MacOSX/Linux, free) (Link)
- **mqtt-spy** (based on Java 8, open source) (Link)
- **MQTT Inspector** (iOS, $1.99) (Link)
- **MyMQTT** (Android, free) (Link)

Some MQTT Broker

- **Eclipse Mosquitto** is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3 (Link)
- **HiveMQ** is now open source, compatible with MQTT 3.1, 3.1.1 and MQTT 5 (Link)
- **Moquette** is a lightweight MQTT broker for the Internet of Things. Simply embeddable in your IoT projects (Link)

The other apps based on MQTT

- Facebook Messenger. Facebook has used aspects of MQTT in Facebook Messenger for online chat. However, it is unclear how much of MQTT is used or for what.
- Amazon Web Services announced *Amazon IoT* based on MQTT in 2015
- Microsoft Azure IoT Hub uses MQTT as its main protocol for telemetry messages

Chapter 5: 6lowPAN - MQTT – CoAP

- Problem definition
- Introduction to IoT Protocols
- 6lowPAN
- Message Queuing Telemetry Transport (MQTT)
- Constrained Application Protocol (CoAP)

CoAP - Overview

- CoAP (Constrained Application Protocol)
- CoAP is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things
- The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation
- One-to-one communication protocol (RFC 7252)

CoAP vs HTTP

- Based on REST model, like HTTP
- CoAP can carry different types of payloads, and can identify which payload type is being used. CoAP integrates with XML, JSON, or any data format of your choice.

CoAP Packet

- CoAP packets are much smaller than HTTP TCP flows. Packets are simple to generate and can be parsed in place without consuming extra RAM in constrained devices.
- CoAP runs over UDP, not TCP. Clients and servers communicate through connectionless datagrams. Retries and reordering are implemented in the application stack.

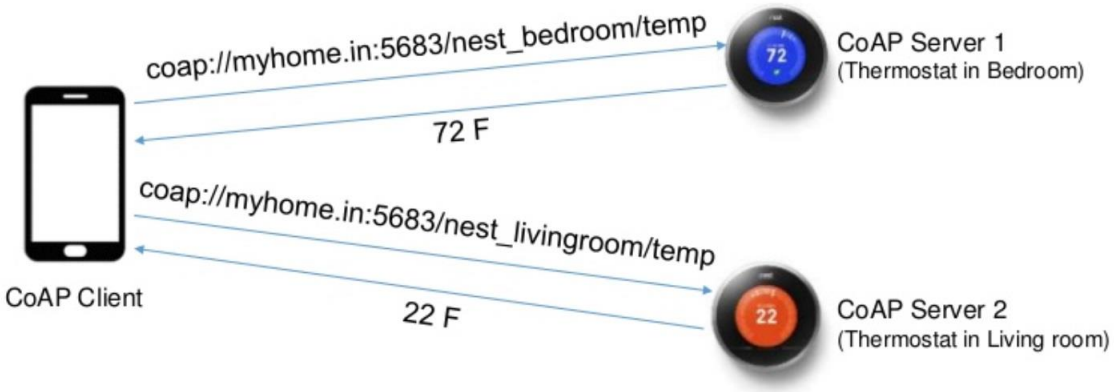
CoAP - QoS

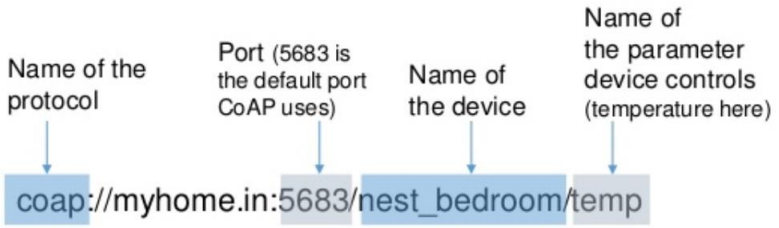
- Requests and response messages may be marked as "*confirmable*" or "*nonconfirmable*".
  - Confirmable messages must be acknowledged by the receiver with an ack packet.
  - Nonconfirmable messages are "fire and forget".

CoAP - Security

- Because CoAP is built on top of UDP not TCP, SSL/TLS are not available to provide security.
- DTLS, Datagram Transport Layer Security provides the same assurances as TLS but for transfers of data over UDP.
- Typically, DTLS capable CoAP devices will support RSA and AES or ECC and AES.

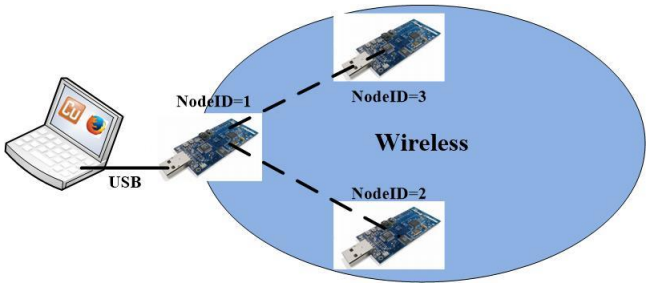
CoAP – Request Response

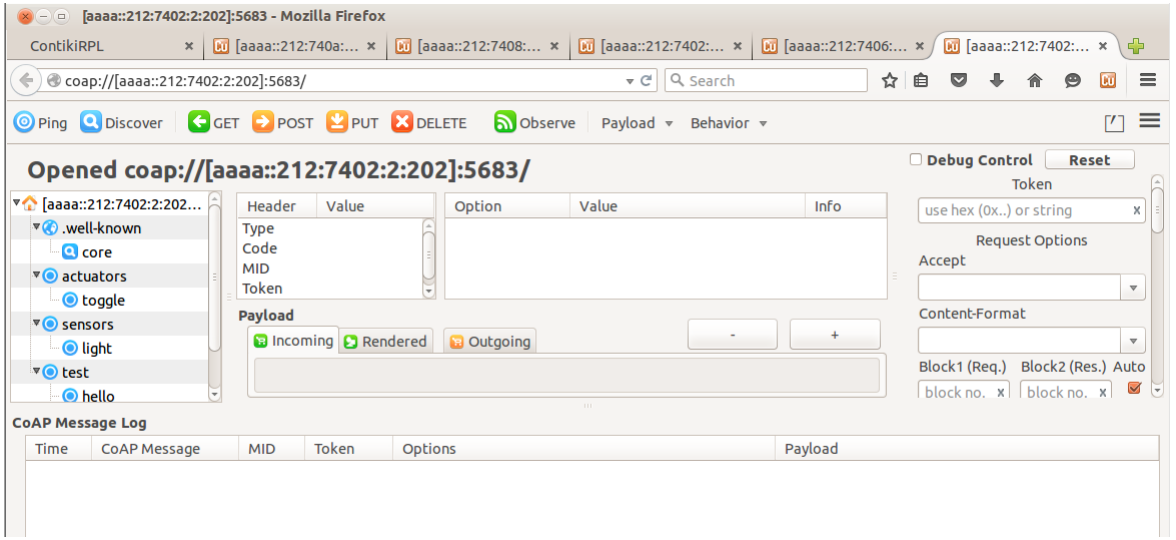




CoAP – A case study

- *ContikiOS/Cooja*
- *Node 1: /contiki-3.0/examples/ipv6/rpl-border-router/*
- *Node 2, 3: /contiki-3.0/examples/er-rest-example*
- *Firefox/Copper (Cu)*





Some CoAP Implementations

- **libcoap** is a C implementation of a lightweight application-protocol for devices that are constrained their resources such as computing power, RF range, memory, bandwidth, or network packet sizes (Link)
- **TinyOS CoapBlip** is used in the TinyOS blip-rpl stack for UDP communication. (Link)
- **FreeCoAP** is An implementation of the CoAP protocol for GNU/Linux (Link)

Summary

- How to run TCP/IP in IoT end devices?
- IoT Protocol stack
- MQTT - overview, Pub/Sub model, QoS
- CoAP - overview, QoS, Security


Q & A