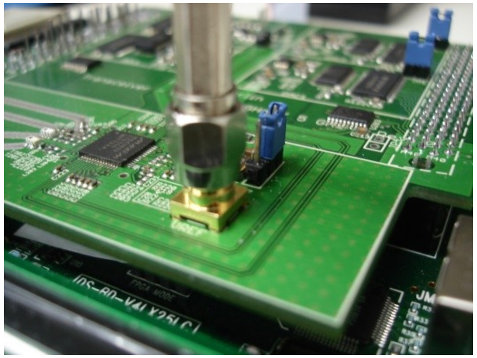
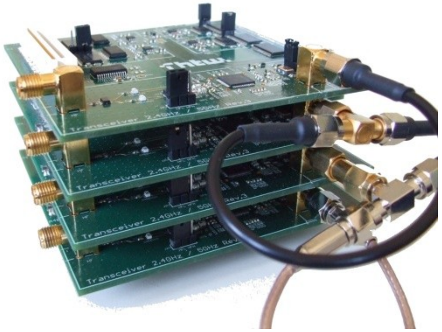
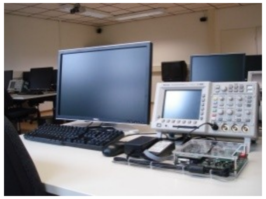
Prof. Dr. Alexander Huhn   HTW – Berlin







Internet of Things

Protocols

Dr. Alexander Huhn · 14.09.22

Content

- **MQTT**
  - **Moquitto – Broker**
  - **Eclipse Paho – client library**
- **CoAP**

MQTT
MQ Telemetry Transport

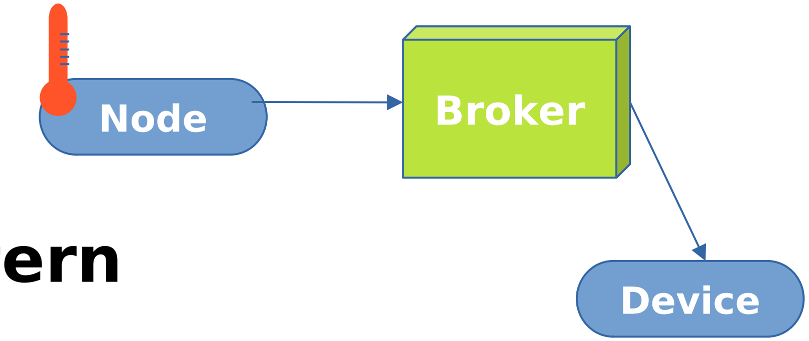
http://docs.oasis-open.org/mqtt/mqtt/

- A Client Server publish/subscribe messaging transport protocol
- Light weight, open, simple, and designed to be easy to implement
- Ideal for use in constrained environments
  - Communication in Machine to Machine (M2M)
  - Internet of Things (IoT)
  - Contexts where a small code footprint is required and/or network bandwidth is at a premium

MQTT
MQ Telemetry Transport

- **Protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bi-directional connections**
- **Features:**
  - **Use of the publish/subscribe message pattern which provides one-to-many message distribution and decoupling of applications**
  - **A messaging transport that is agnostic to the content of the payload.**

MQTT
MQ Telemetry Transport



- **Publish/Subscribe Pattern**
  - **Space decoupling**
    - **Publisher an subscriber do not need to „know each other“ (onnly broker)**
  - **Time decoupling**
    - **No need to be connected at the same time**
  - **Synchronisation decoupling**
    - **Publishers are not blocked while publishing (no need to wait for subscriber to process message)**

MQTT
MQ Telemetry Transport

- **Features:**
  - **Three qualities of service (QoS) for message delivery:**
    - **"At most once"**
    - **"At least once"**
    - **"Exactly once"**

MQTT
MQ Telemetry Transport

- **Features:**
  - **Three qualities of service for message delivery:**
    - **"At most once"**
      - **Messages are delivered according to the best efforts of the operating environment. Message loss can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.**
    - **"At least once"**
    - **"Exactly once"**

MQTT
MQ Telemetry Transport

- **Features:**
  - **Three qualities of service for message delivery:**
    - **"At most once"**
    - **"At least once"**
      - **Messages are assured to arrive but duplicates can occur**
    - **"Exactly once"**

MQTT
MQ Telemetry Transport

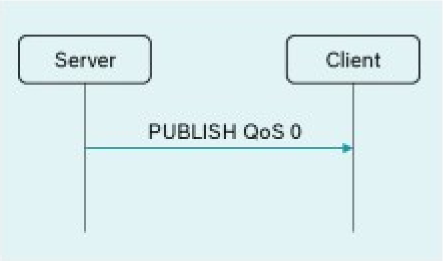
- **Features:**
  - **Three qualities of service for message delivery:**
    - **"At most once"**
    - **"At least once"**
    - **"Exactly once"**
      - **Messages are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.**

MQTT
MQ Telemetry Transport

Quality of Service

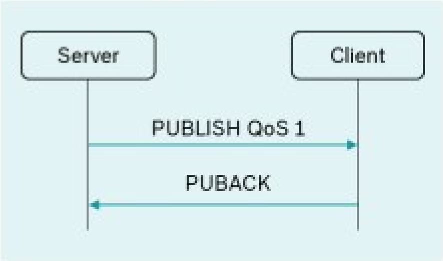
MQTT can guarantee TCP-session-independent reliability

QoS 0



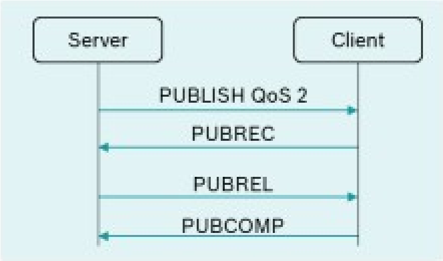
- Fire & forget
- Underlying TCP guarantees
- Useful in stable networks

QoS 1



- Guarantees delivery at least once
- Most applied QoS level
- Duplicates may occur

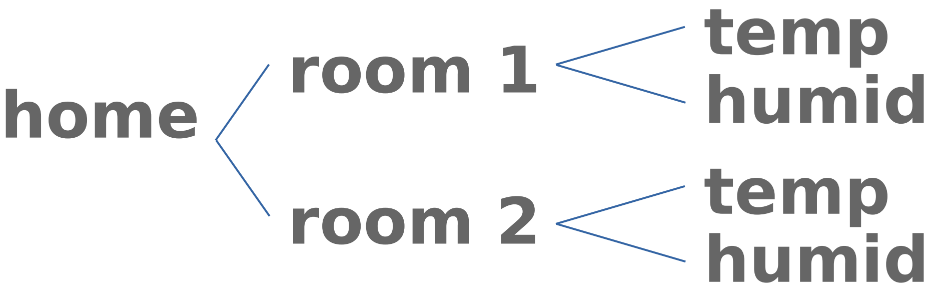
QoS 2



- Guarantees delivery exactly once
- High overhead, slow
- Useful if duplicates would be harmful

MQTT
Topic Structure

- **Hierachical structure**
- **UTF-8 string – case sensitive**
- **Topic wildcards**
  - **Single level +**
  - **Multi level #**



Examples

house/room1/temp   house/+/temp   tempperature of all rooms

house/room1/# all room sub-topics

MQTT
MQ Telemetry Transport

- **Features:**
  - **A small transport overhead and protocol exchanges minimized to reduce network traffic**
  - **A mechanism to notify interested parties when an abnormal disconnection occurs**

MQTT 3.1.1 is also an ISO standard (ISO/IEC 20922)

https://mosquitto.org/

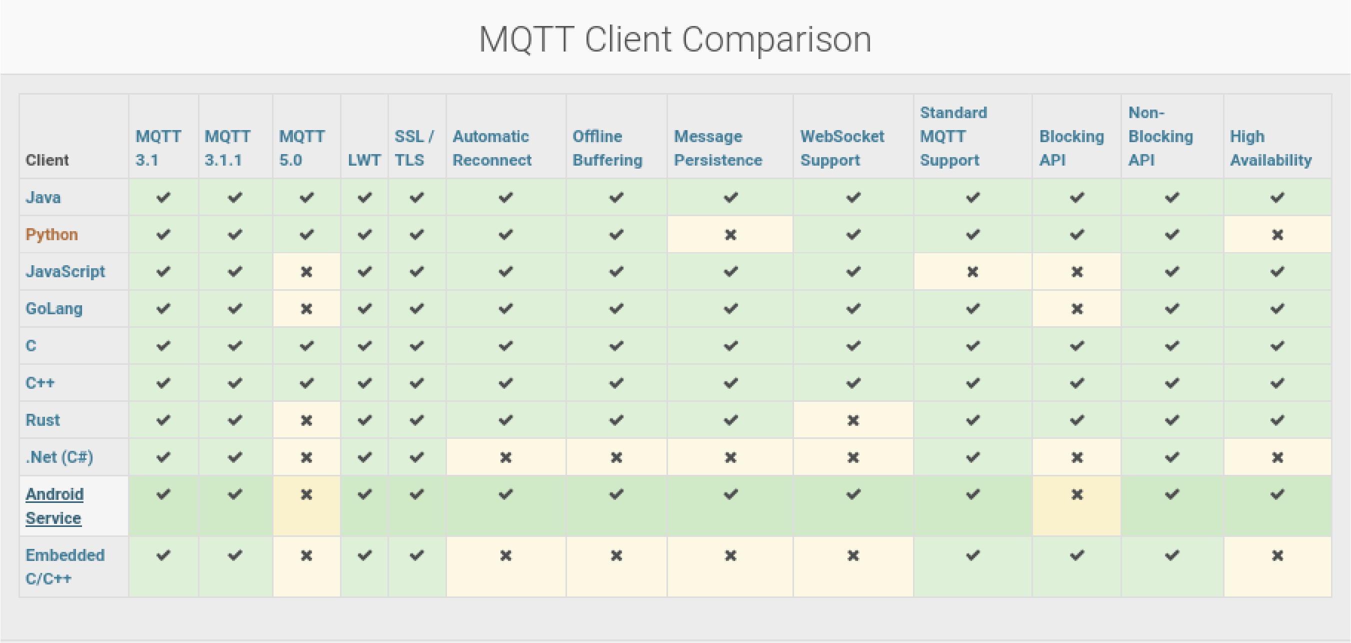
Mosquitto – open source message broker

- **open source (EPL/EDL licensed) message broker**
- **implements the MQTT protocol versions 5.0, 3.1.1 and 3.1**
- **Lightweight**
  - **Suitable for use on all devices from low power single board computers to full servers**

https://www.eclipse.org/paho/

Eclipse Paho – client library

- Eclipse Paho project provides open source implementations of MQTT
  - For a variety of programming languages



MQTT Client Comparison

Client	MQTT 3.1	MQTT 3.1.1	MQTT 5.0	LWT	SSL / TLS	Automatic Reconnect	Offline Buffering	Message Persistence	WebSocket Support	Standard MQTT Support	Blocking API	Non-Blocking API	High Availability
Java	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Python	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
JavaScript	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓
GoLang	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C++	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rust	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
.Net (C#)	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗
Android Service	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Embedded C/C++	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗

References

[mqtt-v5.0]

MQTT Version 5.0. Edited by Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta. 07 March 2019. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Vielen Dank für Ihre Aufmerksamkeit!

RFC 1122

https://www.rfc-editor.org/rfc/rfc1122

Application Layer

The application layer is the top layer of the Internet protocol suite. The Internet suite does not further subdivide the application layer, although some of the Internet application layer protocols do contain some internal sub-layering. The application layer of the Internet suite essentially combines the functions of the top two layers -- Presentation and Application -- of the OSI reference model.

We distinguish two categories of application layer protocols: user protocols that provide service directly to users, and support protocols that provide common system functions. Requirements for user and support protocols will be found in the companion RFC [INTRO:1].

The most common Internet user protocols are:

o Telnet (remote login)
o FTP (file transfer)
o SMTP (electronic mail delivery)

There are a number of other standardized user protocols [INTRO:4] and many private user protocols.

Support protocols, used for host name mapping, booting, and management, include SNMP, BOOTP, RARP, and the Domain Name System (DNS) protocols.

Transport Layer

The transport layer provides end-to-end communication services for applications. There are two primary transport layer protocols at present:

o Transmission Control Protocol (TCP)
o User Datagram Protocol (UDP)

TCP is a reliable connection-oriented transport service that provides end-to-end reliability, resequencing, and flow control. UDP is a connectionless ("datagram") transport service.

Other transport protocols have been developed by the research community, and the set of official Internet transport protocols may be expanded in the future. Transport layer protocols are discussed in Chapter 4.

Internet Layer

All Internet transport protocols use the Internet Protocol (IP) to carry data from source host to destination host. IP is a connectionless or datagram internetwork service, providing no end-to-end delivery guarantees. Thus, IP datagrams may arrive at the destination host damaged, duplicated, out of order, or not at all. The layers above IP are responsible for reliable delivery service when it is required. The IP protocol includes provision for addressing, type-of-service specification, fragmentation and reassembly, and security information.

The datagram or connectionless nature of the IP protocol is a fundamental and characteristic feature of the Internet architecture. Internet IP was the model for the OSI Connectionless Network Protocol [INTRO:12].

ICMP is a control protocol that is considered to be an integral part of IP, although it is architecturally layered upon IP, i.e., it uses IP to carry its data end-to-end just as a transport protocol like TCP or UDP does. ICMP provides error reporting, congestion reporting, and first-hop gateway redirection.

IGMP is an Internet layer protocol used for establishing dynamic host groups for IP multicasting.

The Internet layer protocols IP, ICMP, and IGMP are discussed in Chapter 3.

Link Layer

To communicate on its directly-connected network, a host must implement the communication protocol used to interface to that network. We call this a link layer or media-access layer protocol.

There is a wide variety of link layer protocols, corresponding to the many different types of networks. See Chapter 2.