

Walther- Rathenau- Gewerbeschule Freiburg	<h1 style="margin: 0;">mqtt</h1> MQ Telemetry Transport Topics, Wildcards	Fach: ITS	Gruppe:	
		Dat.: 2. Mai 2019		Seite 1
		Name:		
		Klasse: E3FI1		
		Punkte: /20	Note:	

1 MQTT Topics

Um die Menge an Daten gut organisieren zu können, werden beim Veröffentlichen von Werten sog. *Topics* vergeben. Die Topics sind dabei ähnlich wie die Pfade in einem Dateisystem *hierarchisch* organisiert.

Beispiele:

```

EG/bad/temperatur
EG/bad/luftfeuchte
EG/wohnzimmer/temperatur
EG/kueche/rauchmelder
EG/flur/temperatur
1.OG/flur/temperatur

```

Die Topic-Zeichenketten werden im *variablen Header*, der auf den festen Header folgt, übertragen.

Durch die Topics kann der Broker die eingehenden Nachrichten hierarchisch abspeichern und Clients können durch Angabe eines Topics dann wiederum bestimmte Nachrichten abrufen (abonnieren = subscribe).

1.1 Topic-Wildcards

Möchte man eine ganze Klasse von Topics abonnieren, kann man *Wildcards* einsetzen.

+ Mit dem + - Zeichen kann eine beliebige, einzelne Hierarchieebene angegeben werden. Diese Wildcards werden *Single-Level-Wildcards* genannt.

Beispiel:

```
+/flur/temperatur
```

Mit dem # - Zeichen kann ab einer bestimmten Hierarchie-Ebene alles darunter liegende empfangen werden (*Multi-Level-Wildcards*).

Beispiel:

```
EG/bad/#
```

Das # - Zeichen **muss** dabei am Ende der Topic-Zeichenkette, nach einem Vorwärtsschrägstrich stehen!

1.2 Tipps zu Topics

- Topics sollen **nicht** mit einem Vorwärts-Schrägstrich beginnen. Das ist zwar erlaubt, führt aber nur eine überflüssige, zusätzliche Hierarchieebene ohne weitere Information ein.
- **Niemals Leerzeichen in Topics verwenden.** Das gilt allgemein auch für Dateinamen und Fachleute machen sowas **nie**.
- **Nur ASCII-Zeichen verwenden!** Keine Umlaute, Sonderzeichen oder gar nicht-druckbare UTF8-Zeichencodes! Und auch hier gilt das natürlich auch für Datei- und DNS-Namen.
- Es ist ratsam, einen eindeutigen Client-Bezeichner (z.B. eine Client-ID) in den Topic einzubetten. Das hilft sehr, wenn man herausfinden möchte, wer eine bestimmte Meldung veröffentlicht hat.
- Die Topic-Hierarchienamen sollten so kurz und prägnant wie möglich sein. Das Topic wird im Variablen Header übertragen und sollte kompakt bleiben.
- Bei der Topic-Hierarchie auf spätere Erweiterbarkeit achten.
- Die Topics sollten einen Wert exakt spezifizieren. Also z.B. für jeden Sensor ein eigenes Topic-Level am Ende des Topic-Pfads verwenden und nicht mehrere Sensoren zusammenfassen.

2 Quality-of-Service

Es gibt 3 Qualitätsstufen bei MQTT:

At most once (0) die Nachricht wird vom Client genau einmal gesendet und dann gelöscht. Es findet keine Kontrolle statt, ob die Nachricht korrekt übertragen wurde. Der Empfänger sendet keine Empfangsbestätigung. Da die Nachricht vom Sender nach dem Abschicken gelöscht wird, kann es im Fehlerfall auch keine Übertragungswiederholung geben.

At least once (1) hier sendet der Empfänger eine Bestätigung und der Sender speichert die Nachricht so lange, bis er diese Bestätigung erhält (PUBACK). Eine Nachricht kann also mehrmals übertragen werden.

Exactly once (2) das ist die höchste Qualitätsstufe. Es wird sichergestellt, dass der Empfänger die Nachricht exakt einmal erhält. Erreicht wird dies durch ein 4-Wege-Handshake-Verfahren (siehe Abb. 1).

Die einzelnen Protokoll-Nachrichten beim QoS-Level-2-Handshake sind:

PUBLISH Der Client veröffentlicht die Nachricht mit dem QoS-Level2

PUBREC Der Broker antwortet mit *Publish Received*. Empfängt der Client dieses Publish-Received-Signal nicht, wiederholt er die Sendung der Nachricht mit gesetztem Duplikat-Flag (DUP), bis er schliesslich **PUBREC** erhält.

PUBREL Nachdem der Client das **PUBREC** empfangen hat, kann er bei sich die Nachricht (z.B. Messwert) löschen und quittiert das mit **Publish Release**.

PUBCOMP *Publish Complete*. Diese Nachricht schliesst den Handshake ab. Der Broker kann nun alle zwischengespeicherten Referenzen (z.B. Packet-ID) des Vorgangs löschen.

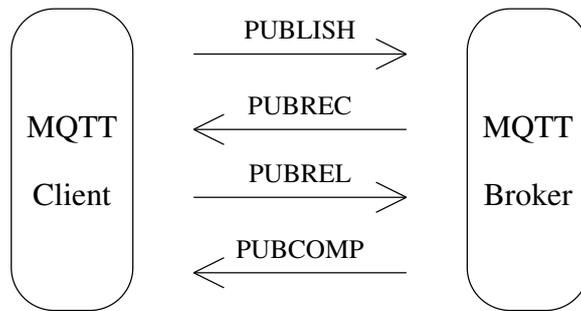


Abbildung 1: 4-Wege-Handshake bei MQTT-QoS-Level-2

2.1 Wann welchen QoS-Level verwenden?

QoS0 QoS0 ist schnell und erzeugt wenig Netzlast. QoS0 kann immer dann verwendet werden, wenn es keine Rolle spielt, ob einige Messwerte verloren gehen.

QoS1 QoS1 stellt sicher, dass mindestens eine Nachricht ankommt, man muss aber auf der Empfängerseite damit umgehen können, dass evtl. Duplikate eintreffen und diese von echten Messwerten unterscheiden können.

QoS2 QoS2 erzeugt keine Duplikate. Sind Duplikate für die Anwendung schädlich, muss man QoS2 verwenden.

3 Retained Messages

Wenn ein Client (Publisher) längere Zeit keine Messwerte sendet, bleiben später hinzukommende Subscriber im Unklaren über den aktuellen Status eines bestimmten Topics.

Der Publisher kann daher auch Messwerte mit gesetztem Retain-Flag veröffentlichen. Diese werden dann vom Broker gespeichert und werden an Subscriber sofort ausgeliefert, auch wenn der Publisher gerade keinen neuen Messwert gesendet hat.

Retained-Meldungen helfen also neuen Abonnenten (Subscriber) sofort nachdem sie ein Topic abonnieren, einen Status zu diesem Topic zu bekommen.

Die Retained-Meldung ist dabei nicht der letzte veröffentlichte Messwert, sondern der letzte veröffentlichte Wert, der mit gesetztem Retain-Flag gesendet wurde.

Löschen kann man eine Retained-Meldung durch Senden einer Nachricht mit

1. gesetztem Retain-Flag
2. entsprechendem Topic
3. leerer Payload zu diesem Topic

Beispiele:

```

pi@raspberrypi:~ $ mosquitto_pub -r -q 2 -h 10.10.0.254 -t qostest -m retainedMessage
pi@raspberrypi:~ $ mosquitto_pub -r -q 2 -h 10.10.0.254 -t qostest -m "
  
```

4 Last Will