

**Documentatie pentru serverul  
MQTT**

**Ceboratos Emil**

**Chiriac Ion**

**Munteanu Letitia-Ioana**

**Grupa 1307A, Facultatea de Automatica si Calculatoare,  
Universitatea Tehnica “Gheorghe Asachi” Iasi**

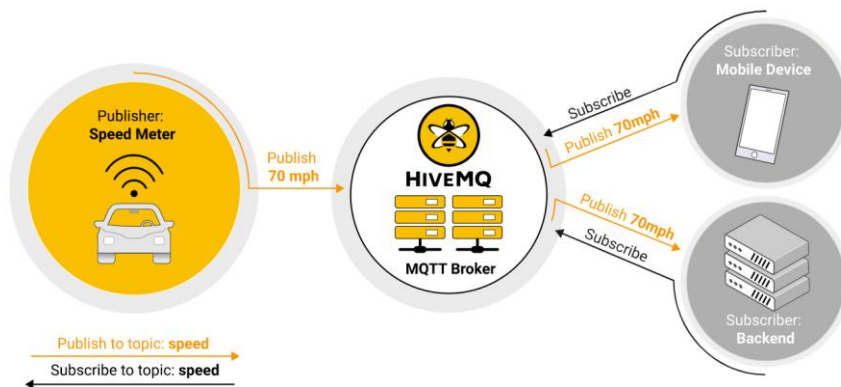
# MQTT – Message Queuing Telemetry Transport

## 1.1 Introducere

MQTT este un protocol rapid, de transmitere a datelor bazat pe modelul Publish/Subscribe. Acesta a fost dezvoltat în 1999 de către Andy Stanford-Clark și Arlen Nipper, conceput astfel încât să fie ușor de implementat, fiind ideal pentru comunicarea Machine-to-Machine (M2M) și IoT (Internet of Things).

## 1.2. Publish & Subscribe

Cel mai important aspect cu privire la acest protocol este decuplarea clientului care trimite un mesaj (Publisher) de cel care primește mesajul (Subscriber) prin intermediul Broker-ului, care joacă un rol important în acest proces.



Decuplarea celor 2 componente cuprinde 3 aspecte generale:

- Spatiul: Clientul de tip Publisher și cel de tip Subscriber nu trebuie să se cunoască, astfel nu există niciun schimb de adrese IP.
- Timpul: Cei 2 clienți nu trebuie să ruleze în același timp.
- Sincronizarea: Operațiile executate de ambii clienți (publicarea unui mesaj sau primirea acestuia) nu trebuie să fie întrerupte.

După cum am spus, Brokerul are un rol important în tot acest proces, întrucât există niște reguli în primirea mesajelor. Nu vrem ca toți subscriberii să primească toate mesajele

publicate, asadar regula este ca fiecare subscriber sa primeasca doar mesajele de interes propriu. Astfel, exista 3 tipuri de filtrare a mesajelor, dupa cum urmeaza:

- SUBJECT-BASED FILTERING
- CONTENT-BASED FILTERING
- TYPE-BASED FILTERING

Cel mai des folosit mod de a filtra mesajele este Subject-Based Filtering care se bazeaza pe un topic. Fiecare Subscriber se aboneaza la un topic de interes iar de aici Broker-ul se asigura ca acesta primeste toate mesajele publicate pentru topicul respectiv.

Content-Based Filtering se refera la filtrarea mesajelor pe baza unui filtru de limbaj in functie de un anumit continut. Continutul mesajului trebuie stiut dinainte si nu poate fi criptat sau schimbat foarte usor.

Type-Based Filtering este folosit atunci cand se utilizeaza limbaje orientate-obiect.

### **Aspecte importante:**

MQTT nu trebuie confundat cu o coada de mesaje din urmatoarele motive:

1. O coada de mesaje stocheaza mesajele pana cand acestea sunt procesate. In acest protocol este posibil ca un mesaj sa nu fie procesat daca niciun client nu se aboneaza la topicul aferent.
2. Intr-o coada obisnuita de mesaje, un mesaj poate fi procesat doar de un client, pe cand protocolul MQTT permite fiecarui client abonat la un anumit topic sa primeasca mesajul dorit chiar daca exista mai multi abonati la acelasi topic.
3. Cozile de mesaje au nume si trebuie create explicit, pe cand topic-urile sunt extrem de flexibile si pot si create din mers.

## **1.3 Topic-urile**

Ce sunt topic-urile ? Sa aprofundam acest subiect.

Topic-urile sunt string-uri pe care Brokerii le folosesc pentru a filtra mesajele pentru fiecare client conectat. Acestea pot avea mai multe nivele, fiind separate printr-un slash (/), nu trebuie sa fie create inaintea utilizarii, Brokerii acceptand orice topic valid.

Pentru a putea primi mesajele dorite, clientii se aboneaza la un anumit topic, caz in care vor primi mesajele corespunzatoare doar acelui topic, sau pot folosi wildecad-uri pentru a se abona la mai multe topic-uri simultan. Wildecad-urile pot fi Single-Level sau Multi-Level.

### **Wildecard-uri Single-Level: +**

Acest tip de wildecard poate inlocui un singur topic. De exemplu in structura urmatoare “AC/ANUL3/MATERII/+/PROIECT” nivelul simbolizat de + poate fi inlocuit cu orice topic.

Un exemplu corect si complet al acestuia ar putea fi:

“AC/ANUL3/MATERII/RCP/PROIECT”. Pentru ca topic-ul sa fie corect, nu avem voie sa modificam topic-urile deja existente sau sa adaugam mai multe topic-uri acolo unde este nevoie: “AC/ANUL3/MATERII/RC/RCP/PROIECT”, “AC/ANUL3/RC/PROIECT”.

### **Wildecard-uri Multi-Level: #**

Acest tip de wildecard poate inlocui mai multe topicuri. De exemplu pentru urmatoarea structura “UNIVERSITATEA/GH.ASACHI/#” pot exista ca si variante corecte urmatoarele:

“UNIVERSITATEA/GH.ASACHI/AC/CTI”,

“UNIVERSITATEA/GH.ASACHI/AC/CTI/SPECIALIZARE/TI”. Varianta

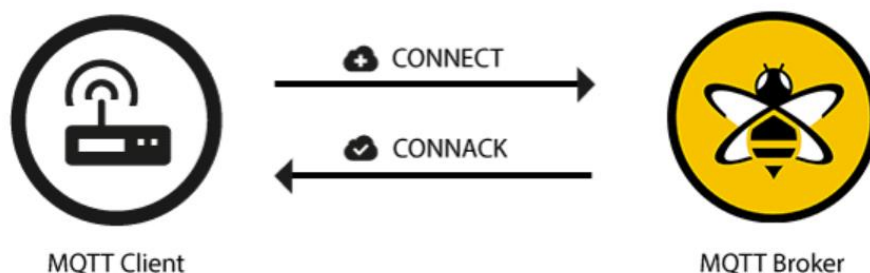
“UNIVERSITATEA/AC/CTI” nu este corecta.

### **Sugestii pentru a scrie corect topic-urile:**


1. Nu se foloseste slash la inceputul topic-ului: “/ac/cti”
2. Nu se folosesc spatii in topic-uri
3. Se folosesc cuvinte scurte si concise
4. De preferat se folosesc caractere ASCII

## **1.4 Client, Broker/Server si stabilirea conexiunii**

Clientul este orice device care ruleaza o librerie MQTT si se conecteaza la un Broker MQTT. Brokerul este responsabil pentru primirea tuturor mesajelor, filtrarea lor, determinarea clientilor abonati si trimiterea mesajelor abonatilor in functie de topic-urile alese de ei.



Pentru a stabili o conexiune, clientul trimite Broker-ului un mesaj de tip CONNECT care contine mai multe campuri. In cazul in care aceste campuri nu respecta specificatiile MQTT, Broker-ul incheie conexiunea.

MQTT-Packet:	
CONNECT	
	
contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

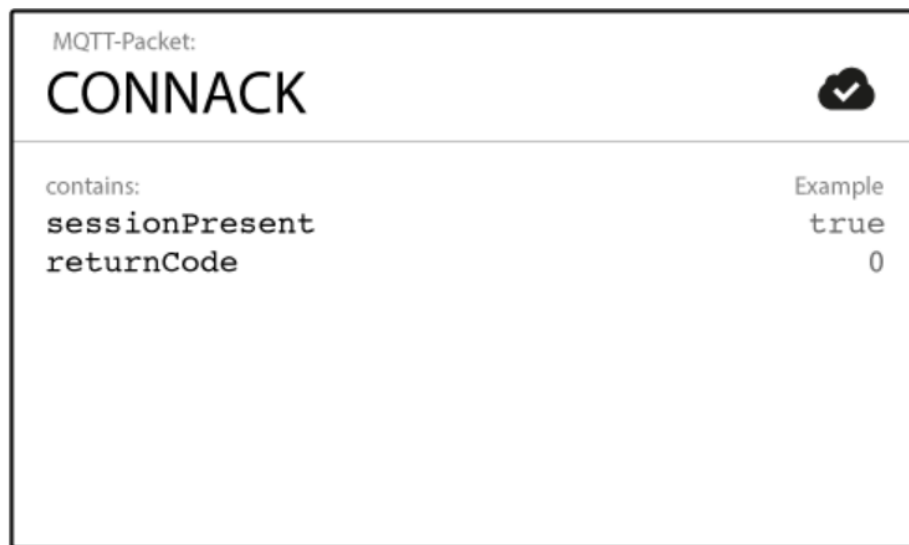
Client ID: identifica fiecare client care se conecteaza la Broker

Clean Session: acest flag stabileste ce tip de conexiune doreste clientul (persistent session non-persistent session).

Last Will Message: acest camp anunta ceilalti clienti cand un client s-a deconectat intr-un mod neasteptat.

Keep Alive: este un interval de timp masurat in secunde care defineste cea mai lunga perioada de timp in care conexiunea intre broker si client este valabila fara ca cel din urma sa trimita broker-ului mesaje

Atunci cand Broker-ul primeste un mesaj de tip CONNECT, este obligat sa raspunda clientului cu un mesaj de tip CONNACK, care, spre deosebire de CONNECT, contine doar 2 campuri.



Session Present: acest flag ii spune clientului daca broker-ul are deja o “persistent session” valabila din interactiunile anterioare cu clientul.

Return Code: ii spune clientului daca incercarea de conectare a avut loc cu succes sau nu.

Return Code	Return Code Response
0	Connection accepted
1	Connection refused, unacceptable protocol version
2	Connection refused, identifier rejected
3	Connection refused, server unavailable
4	Connection refused, bad user name or password
5	Connection refused, not authorized

## 1.5 Publish, Subscribe, Unsubscribe

Publisher: clientul care publica mesaje cat timp este conectat la un Broker.



Topic Name: numele topicului

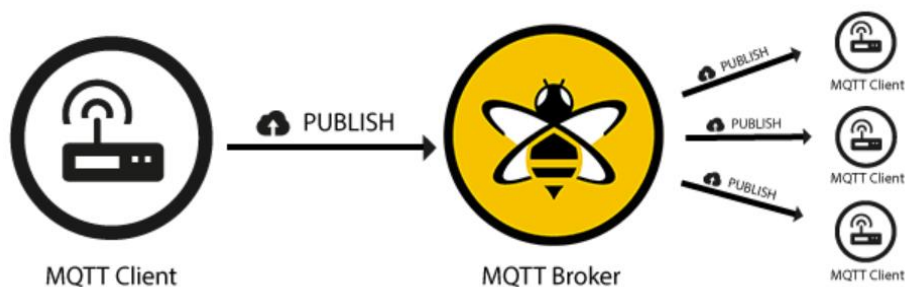
QoS: este un numar ce indica nivelul de Quality of Service (0,1,2)

Retain Flag: acest flag indica daca mesajul este salvat de Broker ca ultima valoare cunoscuta pentru un topic specificat

Payload: acesta este continutul efectiv al mesajului

Packet ID: identificator unic pentru mesaj

DUP Flag: indica daca mesajul este duplicat sau a fost retrimis



Pentru a primi mesajul dorit, clientul trimite un mesaj de tip SUBSCRIBE broker-ului.

MQTT-Packet:	
SUBSCRIBE	
contains:	Example
<b>packetId</b>	4312
<b>qos1</b>	1
<b>topic1</b> } (list of topic + qos)	"topic/1"
<b>qos2</b>	0
<b>topic2</b> }	"topic/2"
...	...

Aceste campuri sunt explicate si mai sus.

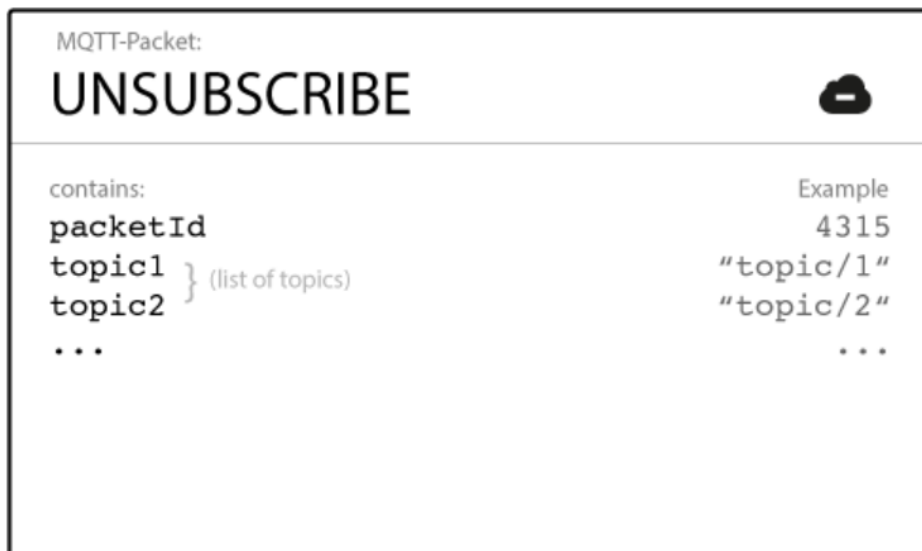
Pentru a confirma fiecare abonare, broker-ul trimite ca raspuns clientului un mesaj de tip SUBACK.

MQTT-Packet:	
SUBACK	
contains:	Example
<b>packetId</b>	4313
<b>returnCode 1</b> ( one returnCode for each	2
<b>returnCode 2</b> topic from SUBSCRIBE,	0
...	...

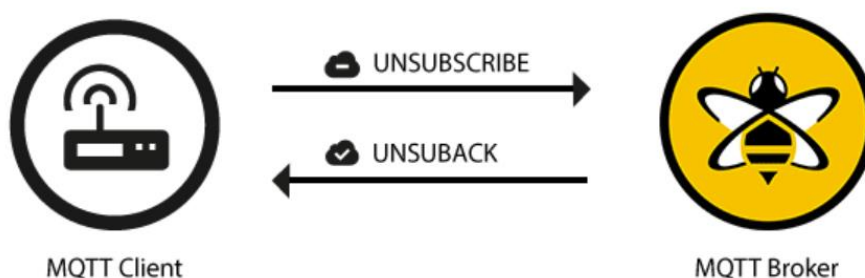
Dupa ce clientul primeste mesajul SUBACK, el va fi abonat la topic-ul ales si va primi mesajele corespunzatoare topic-ului.

Asa cum un client se poate abona la un anumit topic, la fel acesta se poate si dezabona de la acel topic prin intermediul unui mesaj de tip UNSUBSCRIBE care contine identificatorul mesajului (packet ID) si lista de topic-uri de la care doreste sa se dezaboneze.





Pentru a confirma cererea clientului, broker-ul trimite ca raspuns un mesaj de tip **UNSUBACK** care contine doar identificatorul mesajului (packet ID). Dupa ce clientul a primit acest mesaj, acesta este dezabonat de la topic-urile alese.



**Nivelul calitatii serviciului (QoS)** Nivelul calitatii serviciului (QoS) este un acord între expeditorul unui mesaj și receptorul unui mesaj care definește garanția de livrare pentru un mesaj specific. Există 3 niveluri QoS în MQTT:

1. QoS 0 – cel mult o data
2. QoS 1 – cel puțin odata
3. QoS 2 – exact odata

Clientul care publică mesajul către broker definește nivelul QoS al mesajului atunci când trimite mesajul către broker. Brokerul transmite acest mesaj clienților abonați utilizând nivelul QoS pe care fiecare client abonat îl definește în timpul procesului de abonament. Dacă clientul abonat definește un QoS mai mic decât clientul de publicare, brokerul transmite mesajul cu o calitate a serviciului mai mică.

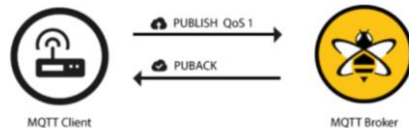
### QoS 0 - cel mult o dată

Nivelul minim QoS este zero. Acest nivel de serviciu garantează o livrare cu cel mai bun efort. Nu există nicio garanție de livrare. Destinatarul nu confirmă primirea mesajului, iar mesajul nu este stocat și re-transmis de către expeditor.



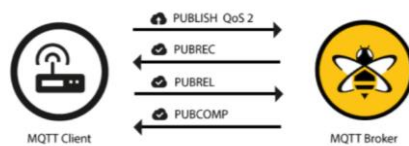
### QoS 1 - cel puțin o dată

QoS nivel 1 garantează că un mesaj este livrat cel puțin o dată receptorului. Expeditorul stochează mesajul până când primește un pachet PUBACK de la receptor care confirmă primirea mesajului. Este posibil ca un mesaj să fie trimis sau livrat de mai multe ori.



### QoS 2 - exact o dată

QoS 2 este cel mai înalt nivel de servicii din MQTT. Acest nivel garantează că fiecare mesaj este primit o singură dată de către destinatarii intenționați. QoS 2 este cel mai sigur și mai lent nivel al calității serviciului. Garanția este asigurată de cel puțin două fluxuri de solicitare / răspuns între expeditor și receptor. Expeditorul și destinatarul utilizează identificatorul de pachet al mesajului PUBLISH original pentru a coordona livrarea mesajului.



### Sesiune persistentă

Pentru a primi mesaje de la un broker MQTT, un client se conectează la broker și creează abonamente la subiectele în care este interesat. Dacă conexiunea dintre client și broker este întreruptă în timpul unei sesiuni non-persistente, aceste subiecte se pierd și clientul trebuie să se aboneze din nou la reconectare. Re-abonarea de fiecare dată când conexiunea este întreruptă este o povară pentru clienții constrânși cu resurse limitate. Pentru a evita această problemă, clientul poate solicita o sesiune persistentă atunci când se conectează la broker. Sesiunile persistente salvează toate informațiile relevante pentru client pe broker.

### Ce este stocat într-o sesiune persistentă?

Într-o sesiune persistentă, brokerul stochează următoarele informații (chiar dacă clientul este offline). Când clientul se reconectează, acesta primește următoarele informații din partea brokerului:

1. Existența unei sesiuni (chiar dacă nu există abonamente).
2. Toate abonamentele clientului.
3. Toate mesajele dintr-un flux pe care clientul nu le-a confirmat încă.
4. Toate mesajele noi pe care clientul le-a ratat în timp ce era offline.
5. Toate mesajele primite de la client care nu sunt încă confirmate complet.

### Cât timp stochează mesaje brokerul?

Brokerul stochează sesiunea până când clienții revin online și primesc mesajul. Totuși, ce se întâmplă dacă un client nu revine online mult timp? De obicei, limita de memorie a sistemului de operare este constrângerea principală pentru stocarea mesajelor. Nu există un răspuns standard pentru acest scenariu. Soluția potrivită depinde de cazul dvs. de utilizare.

### **Mesaje reținute**

Brokerul stochează ultimul mesaj păstrat și QoS corespunzător pentru subiectul respectiv. Fiecare client care se abonează la un model de subiect primește mesajul reținut imediat după abonare. Brokerul stochează un singur mesaj reținut pentru fiecare subiect. Mesajele păstrate ajută clienții nou-abonați să primească o actualizare a stării imediat după abonarea la un subiect. Mesajul reținut elimină așteptarea pentru clienții de publicare pentru a trimite următoarea actualizare. Cu alte cuvinte, un mesaj reținut pe un subiect este ultima valoare bună cunoscută pe care brokerul o trimite noilor abonati.

### **Last Will and Testament**

Fiecare client își poate specifica ultimul mesaj atunci când se conectează la un broker. Brokerul stochează mesajul până când detectează că clientul s-a deconectat fara a trimite un pachet DISCONNECT sau acesta nu reuseste sa comunice intr-o perioada definita. Ca urmare brokerul trimite ultimul mesaj către toți clienții abonați din subiectul ultimului mesaj. Dacă clientul se deconectează cu un mesaj corect DECONNECTARE, brokerul renunță la mesajul LWT stocat.

.

**The Keep Alive** este intervalul maxim de timp care este permis să treacă între punctul în care Clientul termină transmiterea unui pachet de control și punctul în care începe să trimită următorul. Este responsabilitatea clientului să se asigure că intervalul între pachetele de control trimise nu depășește valoarea Keep Alive. În absența trimiterii oricăror alte pachete de control, clientul trebuie să trimită un pachet PINGREQ. Dacă brokerul nu primește un PINGREQ sau orice alt pachet de la un client, brokerul închide conexiunea și trimite ultimul mesaj (dacă clientul a specificat un LWT).