

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DOMENIUL: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

# BROKER MQTT v5

PROIECT - REȚELE DE CALCULATOARE

Profesor coordonator  
Ș.l.dr. Nicolae Botezatu

Student  
Apostol Diana-Florina  
Miron Sebastian

Iași, 2022

# Cuprins

Introducere.....	2
Mecanisme.....	3
Keep Alive .....	3
Last Will.....	3
QoS 0 .....	4
QoS 1 .....	4
QoS 2 .....	5
Stocare sesiuni cu posibilitatea de expirare/ștergere.....	5
Bibliografie .....	6

## Introducere

MQTT este un protocol de transport al mesajelor care se bazează pe TCP/IP și conceptul publish/subscribe. Acesta este proiectat astfel încât să fie accesibil pentru majoritatea calculatoarelor din punct de vedere al resurselor și al lipsei de semnal.

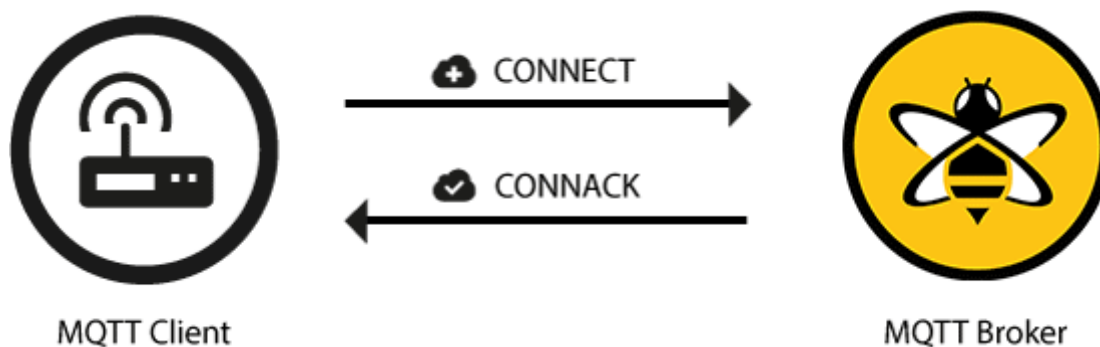
Protocolul a fost inventat în anul 1999 de Andy Stanford-Clark (IBM) și Arlen Nipper (Arcom, acum Cirrus Link) din necesitatea de conectare la conductele de petrol prin satelit cu o pierdere minimă a bateriei și o lățime de bandă cât mai mică.

Cel mai important aspect al acestui protocol este decuplarea totală între publisheri și subscriberi realizată cu ajutorul unui broker.

Broker-ul este inima oricărui protocol publish/subscribe, acesta fiind responsabil pentru primirea tuturor mesajelor, filtrarea mesajelor, determinarea abonaților fiecărui topic și trimiterea mesajului acestor abonați.

O altă responsabilitate a brokerului este autentificarea și autorizarea clienților. Prin urmare, este important ca brokerul să fie foarte scalabil, integrabil în sistemele backend, ușor de monitorizat și rezistent la eșec.

Pentru a iniția o conexiune, clientul trimite un mesaj CONNECT către broker. Brokerul răspunde cu un mesaj CONNACK și un cod de stare. Din acest moment clienții sunt conectați. Brokerul așteaptă primirea de pachete de tip publish ce conțin topic, payload și QoS. În funcție de topic brokerul identifică subscriberii și transmite mai departe mesajul. Orice client poate să fie publisher/subscriber.



# Mecanisme

## 1.Keep Alive

Protocolul TCP care stă la baza MQTT-ului asigură că pachetele sunt transmise într-un mod sigur, în ordine și lipsite de eroare. Cu toate acestea, din când în când se produc desincronizări la nivelul comunicării din cauza întreruperilor de curent sau lipsa semnalului, în aceste situații apar așa-numitele jumătăți de conexiune, ce aduc o funcționalitate eronată a protocolului. MQTT-ul vine cu un mecanism menit să rezolve această problemă numit Keep Alive.

Keep alive asigură că legătura dintre broker și client este încă deschisă și că brokerul și clientul sunt conștienți că sunt conectați. Acest mecanism este implementat cu ajutorul transmiterii pachetelor PINGREQ și PINGRESP.

PINGREQ este un pachet fără payload ce este transmis la un interval predefinit de timp atunci când între client și broker nu există transmitere de pachete. Acesta are rolul de a asigura conexiunea clientului.

PINGRESP este pachetul complementar celui de mai sus, fără payload, și are rolul de a asigura clientul că brokerul încă este conectat. Acesta este transmis de către broker ca un răspuns al pachetului PINGREQ.

În cazul în care între broker și client nu există alt schimb de pachete, iar perioada de timp în care trebuia trimis pachetul PINGREQ expiră, brokerul deconectează automat clientul. În mod asemănător, în cazul în care clientul nu primește răspunsul în timp util este de dorit ca acesta să închidă conexiunea.

## 2.Last Will

Last Will este un mecanism al MQTT-ului ce permite clienților de tip publisher care au fost deconectați în mod forțat, nu prin trimiterea unui pachet de DISCONNECT, să trimită un mesaj predefinit abonaților. Acest mesaj nu are proprietăți specifice, este unul normal cu topic, QoS, payload și poate fi specificat de orice client atunci când se stabilește conexiunea cu brokerul.

Cazurile în care poate apărea folosirea mesajului Last:

- Clientul este deconectat datorită mecanismului Keep Alive.
- Clientul nu transmite pachetul DISCONNECT atunci când închide conexiunea.
- Brokerul închide conexiunea din cauza unei erori de protocol.

### 3.QoS

QoS este un acronim pentru Quality of Service. Acest mecanism tratează modul în care mesajele sunt transmise și primite. În MQTT există trei niveluri de transmitere: QoS 0, QoS 1, QoS 2, fiecare având un mod specific de implementare.

#### QoS 0

Nivelul minim este QoS 0 și se poate defini prin “cel mult o dată”. Acesta ne asigură că mesajul publicat de către publisher este transmis cel mult o dată subscriberilor. În esență, acesta transmite o singură dată pachetul ce conține informația și nu se așteaptă confirmarea primirii.



*Quality of Service level 0: delivery at most once*

#### QoS 1

Nivelul 1 al QoS-ului ne asigură că mesajul transmis ajunge la subscrieri cel puțin o dată. Contrar lui QoS 0 aici se așteaptă confirmarea primirii. În cazul MQTT 3.1.1 dacă nu se primește un mesaj de confirmare într-un timp rezonabil, acesta este retransmis, apărând dezavantajul dublicării mesajelor. În schimb, o dată cu apariția lui MQTT v5 s-a ajuns la concluzia că retransmiterea mesajului nu este cea mai bună variantă deoarece pot apărea aglomerări de pachete. Conceptul de “cel puțin o dată” încă este respectat, dar în loc de retransmiterea mesajului, acum brokerii și clienții retransmit pachete speciale.



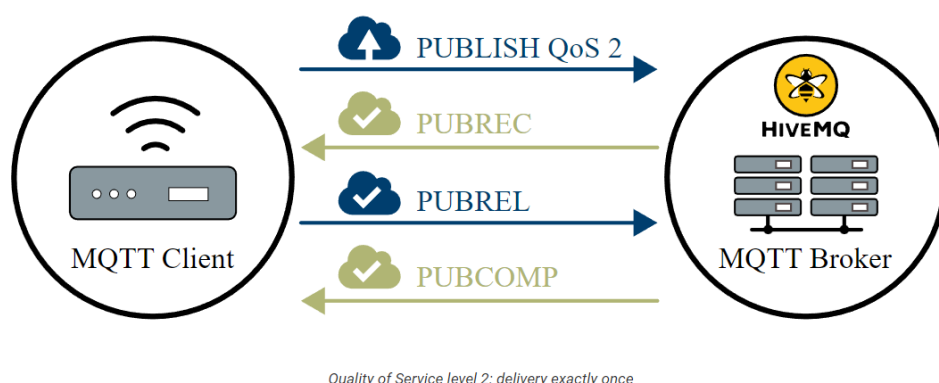
*Quality of Service level 1: delivery at least once*

## QoS 2

QoS 2 este cel mai înalt nivel de serviciu din MQTT. Acest nivel garantează că fiecare mesaj este primit o singură dată de către destinatari. QoS 2 este cel mai sigur, dar și cel mai lent nivel de calitate a serviciului. Când un receptor primește un pachet PUBLISH de la un expeditor, acesta procesează mesajul de publicare și îi confirmă expeditorului cu un pachet PUBREC. Dacă expeditorul nu primește pachetul PUBREC de la receptor, acesta trimite din nou pachetul PUBLISH cu un steag duplicat (DUP) până când primește o confirmare.

Expeditorul stochează pachetul PUBREC de la receptor și răspunde cu un pachet PUBREL.

După ce receptorul primește pachetul PUBREL, acesta răspunde cu un pachet PUBCOMP. Până când receptorul finalizează procesarea și trimite pachetul PUBCOMP înapoi expeditorului, acesta stochează o referință la identificatorul pachetului PUBLISH original. Acest pas este important pentru a evita procesarea mesajului a doua oară.



## Stocare sesiuni cu posibilitatea de expirare/ștergere

În pachetul CONNECT, un client care se conectează poate seta un interval de expirare a sesiunii în secunde. Acest interval definește perioada de timp în care brokerul stochează informațiile despre sesiunea respectivului client, din momentul deconectării acestuia. Când intervalul de expirare a sesiunii este setat la 0 și pachetul CONNECT nu conține o valoare de expirare, informațiile despre sesiune sunt eliminate de broker de îndată ce conexiunea la rețea a clientului se încheie.

## Bibliografie

- <https://www.hivemq.com/mqtt-5/>
- <https://www.hivemq.com/mqtt-essentials/>
- <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>