# Server MQTT(Message Queuing Telemetry Transport)

MQTT este un protocol de comunicare M2M (machine - to - machine) care functioneaza pe modelul publish/subscribe.

* - este un protocol usor de implementat si rapid in transmisia de date;
* - este bazat pe tehnica de mesaj, deci de aici, foarte rapid;
* - pachete de date minimizate, consum redus de retea.

## Functionare MQTT - modelul PUBLISH/SUBSCRIBE

MQTT se bazeaza pe conceptele de clienti si de server.Server-ul este cel responsabil cu gestiunea cererilor clientilor de a trimite sau de a primi date unul de la altul.

Acest model decupleaza clientul care trimite un mesaj (publisher) de clientul/clientii care primesc acele mesaje (subscribers).Conexiunea dintre ei este manipulata de o a treia componenta si anume, broker-ul, a carei informatie de routare este reprezentata de “topic”.Fiecare client care transmite un mesaj trebuia sa includa si topicul mesajului. In acelasi sens, fiecare client in cerere de mesaj se aboneaza la topicul corespunzator.

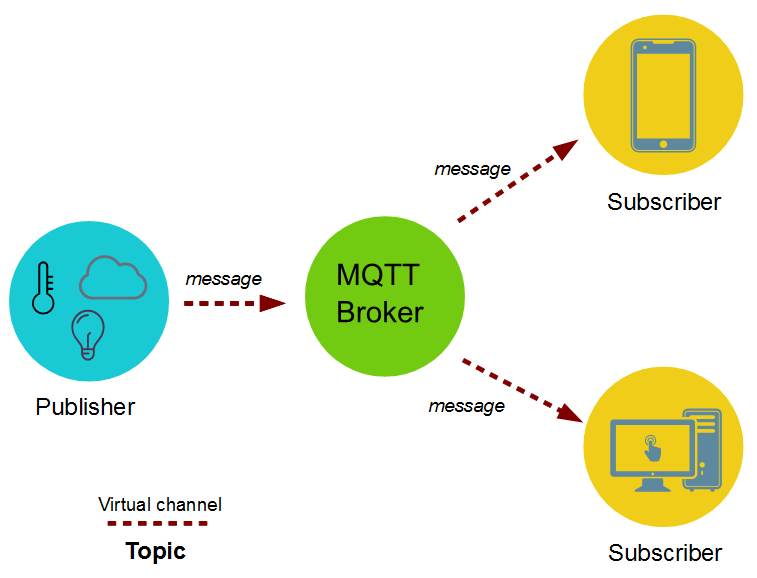
Scopul broker-ului este de a filtra mesajele aparute in sistem si de a le distribui corect catre subscriberi, in functie de topicuri

Astfel:

* cand un dispozitiv (client) doreste sa transmita date catre broker , operatiunea se numeste “publish”;
* cand un dispozitiv (client) doreste sa primeasca date de la broker , operatiunea se numeste “subscribe”;
* la un subiect se pot abona mai multi clienti;
* nu exista legaturi directe intre clienti, putandu-se astfel conecta noi dispozitive fara a incarca structura retelei ( este de ajuns sa fie cunoscute IP-ul si portul broker-ului).Broker-ul si MQTT se comporta ca o interfata comuna .

Cum filtreaza broker-ul mesajele astfel incat fiecare subscriber sa primeasca doar mesajele de interes? Optiuni:

1. Filtrare bazata pe subiect/topic: Clientul se aboneaza la broker pentru topic-uri de interes. De la acest punct incolo, broker-ul  se asigura ca respectivul client va primi toate mesajele publicate pe topic-ul la care s-a abonat. In general, topic-urile sunt string-uri cu o structura ierarhica care permit filtrarea bazata pe un numar limitat de expresii.
2. Filtrare bazata pe continut: Broker-ul filtreaza mesajele pe baza unui filtru de limbaj.
3. Filtrarea bazata pe tip/clasa: Utila atunci cand sunt folosite limbaje orientate-obiect.

**Topics/Subscriptions**

Mesajele in MQTT sunt publicate pe subiecte (topic-uri). Aceste subiecte sunt tratate ca o ierarhie, folosindu-se  caracterul “/” ca separator (asemanator sistemului de fisiere).

Exemplu: mai multe calculatoare isi pot publica informatii despre temperatura hard drive-ului (sensors/PC-NAME/temp/HARD-NAME)

Clientii pot primi mesaje prin creare de abonari. O abonare poate fi la un topic explicit, caz in care doar mesajele de pe acel subiect vor fi primite, sau pot include “wildcards”: “+” si “#”.

* “+” poate fi folosit pentru un singur nivel al ierarhiei:

sensors/+/temp/+

* “#” poate fi folosit pentru toate nivelele ramase din ierarhie:
* Pentru un subiect de tipul “a/b/c/d” se pot folosi urmatoarele tipuri de abonari:
* a/# ; a/b/# ; a/b/c/ # ; +/b/c/#

QoS - Quality of Service

Nivelul QoS este un aranjament intre cel care trimite mesajul si cel care il primeste pentru a defini garantia transmiterii unui mesaj specific.

Exista 3 niveluri:

* -At most once (0);
* -At least once(1);
* -Exactly once(2).

Cand vorbim despre QoS in MQTT ,trebuie luate in considerare 2 parti ale trimiterii:

1.Livrarea mesajului de la clientul care publica catre broker.

2.Livrarea mesajului de la broker catre clientul abonat.

Descrierea celor 3 niveluri:

1.QoS 0: “fire and forget” –nu exista garantia trimiterii.mesajul nu este stocat si retransmis

2.QoS 1:garanteaza ca mesajul este trimis macar o data catre receiver.Transmitatorul stocheaza mesajul pana cand primeste un pachet PUBACK de la receptor cum ca a primit mesajul.Este posibil ca mesajul sa fie trimis de mai multe ori.

3.QoS 2:cel mai inalt nivel se serviciu in MQTT.Acest nivel garanteaza ca fiecare mesaj este primit o singura data decatre recipient.Este cel mai sigur si incet QoS.

**Despre TCP/IP(Transmission Control Protocol / Internet Protocol)**

Modelul de referinta TCP/IP reprezinta cel mai flexibil mod de transport disponibil si permite computerelor din intreaga lume, ruland sisteme de operare complet diferite, sa comunice intre ele.

Standardul TCP/IP este folosit in acest moment pentru transmisiile de date din cea mai mare retea existenta - Internetul.

TCP/IP este un set de protocoale stabil, bine definit si complet care asigura transferul pachetelor de date cu o rata foarte mica de eroare printr-o reţea neomogena de calculatoare. TCP/IP este o suita de protocoale, dintre care cele mai importante sunt TCP şi IP.

**Conexiunea MQTT**

Protocolul MQTT este bazat pe TCP/IP.

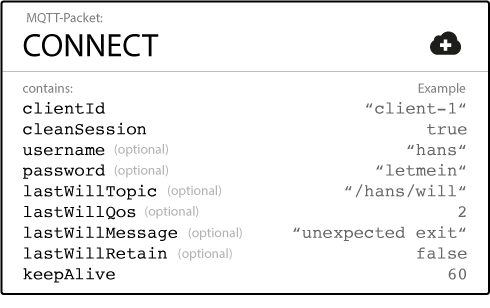
Pentru a initia conexiunea, clientul trimite un mesaj “CONNECT” broker-ului. Acesta raspunde cu un mesaj “CONNACK” impreuna cu un cod status si mentine conectat clientul pana cand acesta trimite o comanda de deconectare sau apare o intrerupere.

Serverul transmite mesajele pe masura ce informatiile sunt actualizate (de aici necesitatea mentinerii conexiunii), iar in cazul unei intreruperi broker-ul MQTT pastreaza datele si le transmite cand conexiunea este restabilita.

**MQTT CONNECT command message**

Clientul poate trimite pachetul CONNECT o singura data. Broker-ul va detecta cazul in care acesta trimite un al doilea pachet de conectare ca o violare de protocol si va deconecta clientul.

  De asemenea, daca mesajul CONNECT nu corespunde specificatiei MQTT sau intervalul de timp intre deschiderea socket-ului si transmiterea mesajului de conectare este mare, broker-ul inchide conexiunea.



ClientId - broker-ul identifica clientul (id ul trebuie sa fie unic) si starea curenta a acestuia;

CleanSession - un flag care specifica daca respectivul client doreste o sesiune “persistenta” (cleanSession = false daca broker-ul pastreaza toate abonarile si mesajele pierdute pentru un client cu QoS de nivel 1 sau 2);

Will Message - face parte din Last Will and Testament (LWT) feature MQTT, broker-ul notificand o desconectare neanuntata cu acest mesaj din partea clientului;

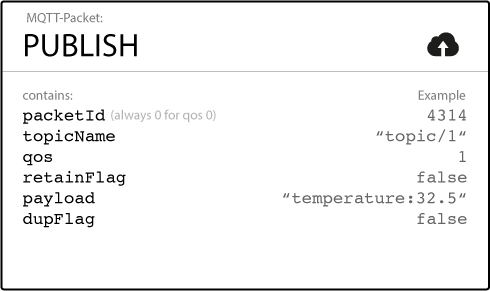
KeepAlive - interval de timp specificat de client care defineste cea mai lunga perioada in care nu sunt transmise mesaje

**MQTT CONNACK message**

Cand un broker primeste un mesaj CONNECT, este obligat sa raspunda cu un mesaj CONNACK, care contine:

* Session present flag - specifica daca exista o sesiune “persistenta” valabila din interactiuni precedente cu clientul;
* Connect acknowlegde flag - contine un return code care transmite clientului daca s-a realizat conexiunea cu succes.

**MQTT PUBLISH message**



RetainFlag - defineste daca mesajul trebuie salvat de catre broker ca ultima valoare cunoscuta pentru un topic specificat. Cand un nou client se aboneaza la topic, acesta primeste ultimul mesaj retinut;

Payload - continutul propriu-zis al mesajului ( MQTT este data-agnostic, putandu-se trimite imagini, text in orice encoding, encrypted data etc);

Cand un client transmite mesajul catre broker pentru publicare, acesta citeste mesajul, il recunoaste ( corespunzator nivelului QoS) si il proceseaza - implicit determina ce clienti sunt abonati la topic si transmite acestora respectivul mesaj.Clientul care publica mesajul nu primeste niciun raspuns in legatura cu cei abonati la topic.

**MQTT SUBSCRIBE message**

Packet Identifier - identifica unic un mesaj;

List of Subscriptions - un mesaj SUBSCRIBE poate contine multiple abonari al clientului. Fiecare abonare este compusa din topic si un nivel QoS.

**MQTT SUBACK message**

Pentru a confirma fiecare abonare, broker-ul trimite un SUBACK mesaj de confirmare clientului. Acest mesaj contine packetId-ul mesajului de abonare original si o lista de coduri de returnare.

