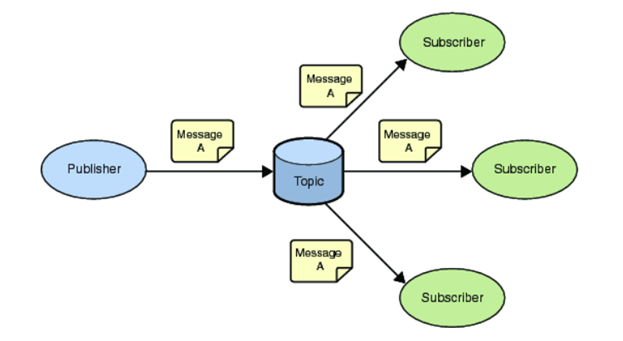
**Message Queuing Telemetry Transport**

**MQTT server**

   Mqtt este un standard de comunicatie prin internet care defineste unul dintre cele mai importante protocoale de transport al mesajelor prin internet intre server si clienti. Acesta  a fost inventat si dezvoltat in 1999 de IBM drept un protocol simplu si flexibil, necesitand un minim de resurse, util pentru conexiunile la distanta si aplicatiile ce fac parte din “Internet of Things”, fie ca sunt automatizari de consum (Consumer IoT) sau automatizari industriale(IIot).

  In 2014 a fost declarat ca standard de OASIS(Organization for the Advancement of Structured Information Standards), iar in 2016 ca  standard ISO(International Organizaion of Standardization).

       Modelul descris de MQTT se bazeaza pe conexiunea dintre server(“broker”) si client, acesta din urma isi publica mesajele cu o eticheta numica “topic”. De asemenea, alti clienti se aboneaza ,”subscribe” la server la anumite topicuri. Brokerul are misiunea de a trimite mesaje unui anumit topic clientilor ce s-au abonat la acel topic.



Un protocol de mesagerie “Publisher- Subscriber” permite publicarea unui mesaj o singură dată și primirea acestuia de  mai mulți consumatori (aplicații /dispozitive) oferind decuplarea între producător și consumatori.

Informațiile sunt organizate într-o ierarhie de topicuri. Atunci când un editor(publisher) are un nou element de date de distribuit, acesta trimite un mesaj de control cu ​​datele către brokerul conectat. Brokerul distribuie apoi informațiile tuturor clienților care s-au abonat(“make a subscription”) la acel topic.

Astfel un server / broker de mesaje “matches publications to subscriptions” in următorul mod:

 • Dacă nu se potrivește, mesajul este eliminat

 • Dacă este una sau mai multe potriviri, mesajul este livrat fiecărui abonat / consumator.

Publisherul nu trebuie să dețină date despre numărul sau locațiile abonaților, iar abonații, la rândul lor, nu trebuie să fie configurați cu niciun fel de date despre publisheri.

Un abonament(„a subscription”) poate fi durabil sau nedurabil:  
 - Durabil:Conform abonamentului(subscription), un broker va transmite mesajele potrivite către abonat,imediat dacă abonatul

este conectat ,dacă abonatul nu este conectat, mesajele sunt stocate pe server / broker până la următoarea dată în care abonatul se conectează .

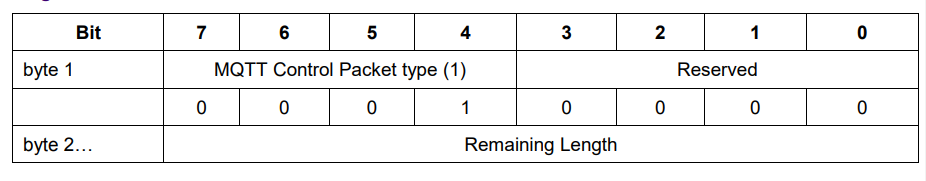
-Nedurabil: durata de viață a abonamentului(subscription) este aceeași cu cea a abonatului conectat la server / broker.

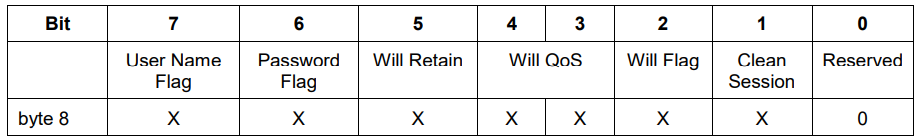
Există patru pachete de control MQTT primare pe care un client și un server le pot folosi pentru a comunica:

-Conectare (“Connect**”**)

— Primul pachet trimis de la client la server trebuie să fie un pachet  de conectare pentru a stabili o conexiune.

Serverul TREBUIE să valideze faptul că “reserved flag” din pachetul de control  
CONNECT este setat la zero  astfel deconectează clientul dacă nu este zero.





Pentru a primi mesaje de la un broker MQTT, un client se conectează la broker și creează abonamente la topicurile de 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. Id-ul clientului oferit atunci când stabilește conexiunea cu brokerul identifică sesiunea

Când clientul se conectează la broker, acesta poate solicita

o sesiune persistentă. Clientul folosește un flag “cleanSession” pentru a spune

brokerului de ce tip de sesiune are nevoie: Când flagul “cleanSession” este setat

pe ‘1’, clientul nu dorește o sesiune persistentă. Dacă clientul se

deconectează din orice motiv, toate informațiile și mesajele aflate în coadă

dintr-o sesiune persistentă anterioară se pierd. Când flagul “cleanSession” este

setat pe ‘0’, brokerul creează o sesiune persistentă pentru client. Toate

informațiile și mesajele sunt păstrate până la următoarea dată când clientul

solicită o “cleanSession”. Dacă flagul “cleanSession” este setat pe ‘0’și

brokerul are deja o sesiune disponibilă pentru client, acesta folosește sesiunea existentă și livrează mesajele la coadă anterior clientului.

-Deconectare(“Disconnect**”)** - Pachetul final trimis de la client la server care indică de ce conexiunea este închisă.

După trimiterea unui pachet “DICONNECT” Clientul:

· TREBUIE să închidă conexiunea la rețea

· NU TREBUIE să mai trimit pachete de control pe acea conexiune de rețea

La primirea DECONECTĂRII serverului:

· TREBUIE să renunțe la orice „Will message”  asociat cu conexiunea curentă fără a-l publica.

· TREBUIE să închidă conexiunea la rețea dacă Clientul nu a făcut deja acest lucru.

Serverul TREBUIE să valideze faptul că biții rezervați(„reserved”) sunt setați la zero și deconectează Clientul dacă nu sunt zero.

-Abonare(“Subscribe”**)**

— Un pachet de abonare este întotdeauna trimis de la client la server pentru a crea unul sau mai multe abonamente la topicuri.

-Publicare(“Publish”)

— Un pachet de publicare poate fi trimis de la client la

server, transportând de obicei un mesaj de aplicație sau de la server la un client care s-a abonat la subiectul respectiv.

Cerintele proiectului:

**Listă de topic-uri configurabilă**(creare/stergere din GUI, fisier config.).

**Tratarea clienților care vor să se aboneze la un topic inexistent.** Clientii vor primi un mesaj corespunzator.

**Vizualizare clienți conectați și abonați, deconectare forțată client.** Se va afisa lista clientilor salvati drept “conectati/abonati”.

**Vizualizarea istoricului pentru ultimele 10 valori publicate/topic.** Se va afisa ultimele topicuri publicate.

**Autentificare clienți.** Clientii se pot autentifica printr-un username/email si o parola.

**Implementare mecanism KeepAlive**.

Keep Alive este un interval de timp măsurat în secunde.

Exprimat ca un cuvânt pe 16 biți, este maxim 539 intervale de timp permise să

treacă între momentul  în care clientul termină transmiterea unui mesaj și momentul în care începe să trimită următorul.

Dacă valoarea Keep Alive este diferită de zero și Serverul nu primește un pachet de control de la client în decurs de o perioadă și jumătate de timp Keep Alive, TREBUIE să deconecteze conexiunea de rețea-Client ca și cum rețeaua ar fi eșuat.

O valoare Keep Alive de zero (0) are ca efect oprirea mecanismului Keep Alive. Aceasta înseamnă că Serverul nu este obligat să deconecteze Clientul din motive de inactivitate,dar o poate face pentru că îi este permis să deconecteze un client pe care îl consideră inactiv sau care nu răspunde  în orice moment, indiferent de valoarea Keep Alive furnizată de clientul respectiv.

**Implementare QoS 0,1,2.**

Quality of Service 0 : publicarea mesajelor fara asteptarea unui feedback din partea serverului.

  Quality of Service 1 : publicarea mesajului in mod repetat pana la o confirmare a serverului.

  Quality of Service 2 : publicarea mesajului, asteptarea confirmarii primirii mesajului, confirmarea ca mesajul poate fi procesat , iar apoi confimarea incheierii transmisiunii.

**Implementare mecanism LastWill**

În MQTT,  se utilizează caracteristica ” Last Will and Testament”(LWT) pentru a anunța alți clienți despre un client deconectat. Fiecare client își poate specifica ultimul mesaj de testament atunci când se conectează la un broker. Ultimul mesaj de testare este un mesaj MQTT normal cu un subiect, flagul mesajului reținut, QoS și ”payload”-ul respectiv. Brokerul stochează mesajul până când detectează că clientul s-a deconectat forțat. Ca răspuns la deconectarea lipsită de grație, brokerul trimite ultimul mesaj către toți clienții abonați la ultimul topic. Dacă clientul se deconectează grațios cu un mesaj corect DECONECTARE, brokerul renunță la mesajul LWT stocat.