

Universitatea Tehnic

˘

a "Gheorghe Asachi" din Ias

,

i

Facultatea de Automatic

˘

as

,

i Calculatoare

Retele de Calculatoare - Proiect

MQTTv5 - Client

ECHIPA: echipa-6˘

Daian Tudor

Grupa 1310A

Costincianu Diana Isabela

Grupa 1310A

1. Descrierea proiectului:

MQTT v5 este un protocol de retea usor, bazat pe paradigma publisher/subscriber, de la masina la masina pentru serviciul de coada de mesaje/cozi de asteptare a mesajelor. Prin sistemul de publicare/abonare se pot publica si primi mesaje in calitate de client. Este conceput pentru conexiuni cu locatii la distanta care au dispozitive cu constrangeri de resurse sau latime de banda limitata, cum ar fi Internetul lucrurilor (Internet of things - IoT). Trebuie sa ruleze printr-un protocol de transport care ofera conexiuni ordonate, fara pierderi, bidirectionale - de obicei, TCP/IP.

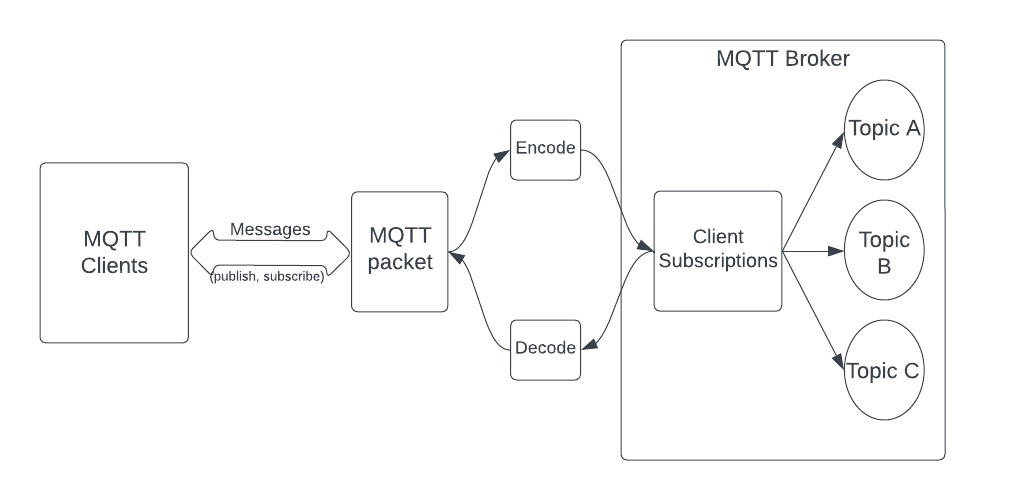


Figure 1: Diagrama functionalitatii MQTT Client + Broker

1. Caracteristici ale MQTT:

MQTT are cateva caracteristici unice care se regasesc cu greu in alte protocoale. Unele dintre acestea sunt:

(LateX template @ NeurIPS 2019)

* + Este un protocol masina la masina ce poate asigura comunicarea intre dispozitive.
  + Este conceput ca un protocol de mesagerie simplu si usor, care utilizeaza un sistem de publicare/abonare pentru a face schimb de informatii intre client si server.
  + Nu necesita ca atat clientul, cat si serverul sa stabileasca o conexiune in acelasi timp.
  + Ofera o transmisie mai rapida a datelor, cum ar fi modul in care WhatsApp/Messenger ofera o livrare mai rapida. Este un protocol de mesagerie in timp real.
  + Permite clientilor sa se aboneze la o selectie restransa de subiecte, astfel incat sa poata primi informatiile pe care le cauta.

1. Analiza proiectului:

In MQTT, abonatul si publisher-ul sunt cele doua roluri ale unui client. Clientii se aboneaza la subiecte pentru a le publica si a primi mesaje. Un dispozitiv este client daca deschide conexiunea de retea la serever, publica mesaje pe care alti clienti doresc sa le vada, se aboneaza la mesajele pe care este interesat sa le primeasca se dezaboneaza la mesajele pe care nu este interesat sa le primeasca s, i inchide conexiunea cu serverul.

Client-ul poate trimite pachete catre server (de tipul connect, publish, subscribe, etc.), iar in functie de QoS-ul(quality of service - Figura 1) atasat pachetului, acesta poate face un schimb cu broker-ul de alte pachete specifice(puback, pubrec, pubrel). Aceste pachete reprezinta o insiruire de octeti, codificati intr-un anumit mod pentru a putea fi decodificate de catre broker si vice versa.

Dupa ce clientul a trimis un pachet de tip CONNECT, acesta trebuie sa astepte sa primeasca un pachet CONNACK de la Broker (cu reason code-ul 0x00 pentru conectare reusita), singurele pachete ce pot fi trimise in acest timp putand fi de tip DISCONNECT sau AUTH. In cazul trimiterii unui pachet de tip SUBSCRIBE, clientul trebuie sa astepte de la Broker un pachet de tip SUBACK. Pentru pachetele de tip PUBLISH, in functie de codul atasat ce descrie nivelul de QoS, clientul si Broker-ul vor face un schimb de diferite pachete (PUBACK, PUBREC, PUBREL, PUBCOMP) pentru confirmarea trimiterii pachetelor. Logica schimbului de pachete pentru QoS-uri de nivel 1 si 2 se poate observa in figura 3 si 4.

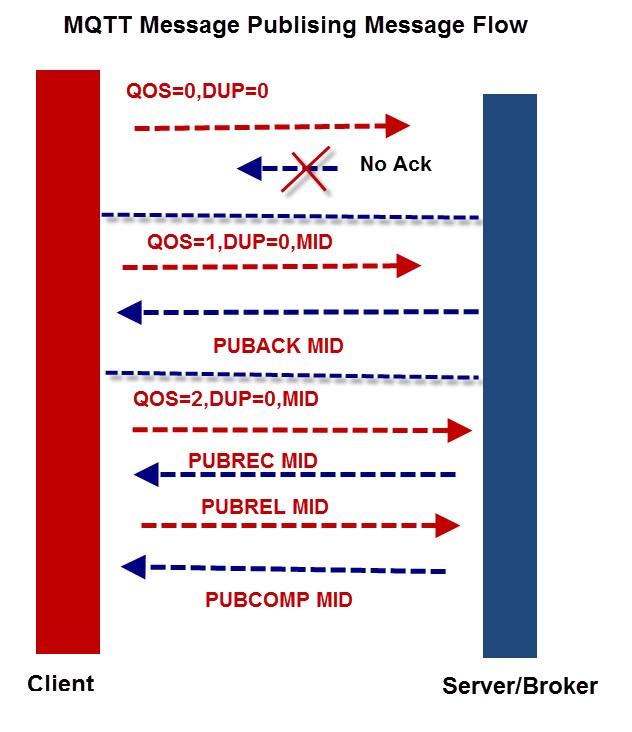


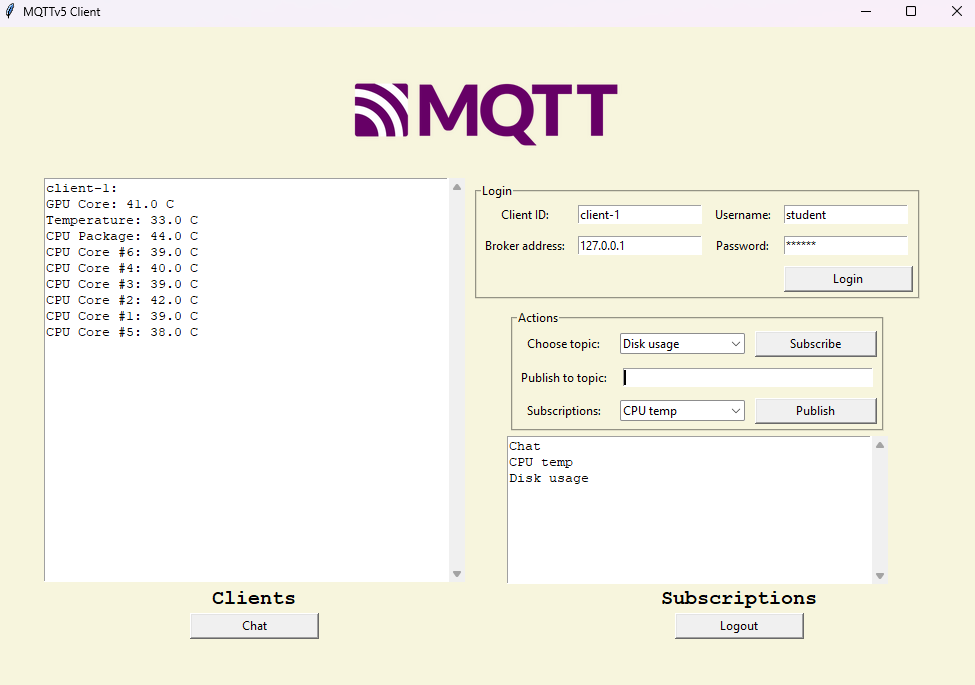
Figure 2: Transpmiterea de pachete in functie de nivelul QoS

In codificarea pachetului de CONNECT, clientul dispune de un mechanism "Keep Alive" care necesita un schimb periodic de pachete (PINGREQ - din partea clientului si PINGRESP - din partea Broker-ului), odata la 5 secunde pentru a valida daca clientul (sau Broker-ul) sunt inca "in viata".

Pachetele sunt alcatuite din:

* Fixed header: primul octet reprezinta tipul pachetului si flag-uri specifice, apoi inca unul sau doi octeti ce reprezinta dimensiunea ramasa a pachetului.
* Variable header: Acesta este diferit pentru fiecare pachet si poate contine flag-uri si date despre pachet.
* Payload: Acesta poate sau nu sa lipseasca din componenta pachetului in functie de tipul acestuia (necesar pentru pachetele connect, subscribe, unsubscribe, suback si unsuback si optional pentru publish). Daca payload-ul este prezent, acesta poate contine date care sunt precedate pe doi octeti de lungimea lor inainte de a fi adaugate propriu-zis in componenta pachetului.

Aceste pachete, prin constructia lor urmand o anumita structura se realizeaza codificarea urmand sa fie decodificate de catre broker. Totodata, pachetele primite de la broker trebuie decodificate de catre client pentru a fi interpretate.



A screenshot of a computer

Description automatically generated

In interfata (GUI) a aplicatiei, utilizatorul se poate loga cu username si parola pentru a avea acces la functionalitatile aplicatiei si poate alege topic-uri la care sa dea subscribe. Totodata acesta poate da publish la anumite informatii pentru ca alti clienti sa poata da subscribe pentru a obtine datele publicate. Utilizatorul dispune de 3 topicuri, acestea fiind topicul de chat, in care utilizatorii pot trimite mesaje broker-ului ca apoi sa fie afisate in fereastra de chat prefixate de id-ul clientului, un topic ce monitorizeaza temperatura nucleelor din procesor si un topic ce afiseaza capacitatea de stocare libera din drive-ul curent. Pentru a afisa temperatura nucleelor este nevoie de rularea pe fundal a aplicatiei open hardware monitor

La delogarea unui utilizator sau la pierderea conexiunii cu acesta, toti clientii vor primi un mesaj de tip LastWill ce ii informeaza ca utilizatorul respectiv a fost delogat sau ca s-a pierdut conexiunea cu acesta.

Aplicatia functioneaza pe trei thread-uri:

* Thread-ul principal (pentru main loop) pe baza caruia functioneaza interfata grafica.
* Thread-ul pentru trimiterea pachetelor de tip ping pentru facilitarea mecanismului de keep alive
* Thread-ul pentru functiile de send si receive.

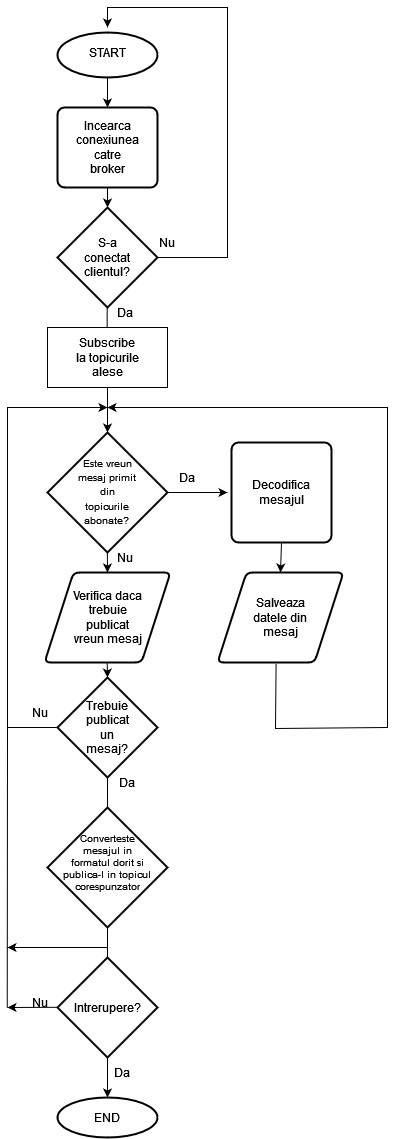


Figure 5: Diagrama functionalitatii clientului