

TP : Le protocole MQTT

J.Gaber

Janvier 2021

gaber@ieee.org

MQTT (Message Queuing Telemetry Transport) est un protocole réseau TCP/IP adapté à l'envoi et la réception de messages en flux continu (streaming). C'est un standard OASIS.

MQTT possède des ports réservés le port 1883 (TCP, UDP) et le port 8883 pour la version sécurisée MQTTS (SSL, TLS).

11. Consulter la spécification complète de MQTT sur le site de l'OASIS ? (donnez le(s) lien(s) visité(s)) ?
12. Rappeler l'architecture de base de MQTT ? (Donnez une figure illustrative et les définitions des termes : Broker (serveur MQTT), producers, consumers, topic) ?
13. Rappeler le principe de fonctionnement ?
14. Quelle est le rôle des jokers (wildcards) ? (Donnez des exemples)
15. Donnez les noms de quelques brokers (i.e., courtiers) les plus populaires ?
16. Quels sont les différents niveaux de la QoS de MQTT ?

Mosquitto

Dans ce qui suit, nous allons nous intéresser au Broker Mosquitto.

Installation

17. Procédez tout d'abord à la mise à jour des paquets de votre système et l'installation des versions les plus récentes de tous les paquets présents (donnez les commandes correspondantes à invoquer dans un terminal) ?
18. En cas de besoin, lancez la commande : `sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa`. Quel est le rôle de cette commande ?
19. Installez ensuite Mosquitto avec la commande `$ sudo apt-get install mosquitto` ?
20. Vérifier que le serveur fonctionne à l'aide de la commande `systemctl` ? (rappeler le rôle de cette commande en consultant sa page man).

Une fois le serveur est installé et en cours d'exécution, nous avons besoin de clients

Installation de clients

21. Installer des clients à l'aide de la commande : `sudo apt install mosquitto-clients`

Pour s'abonner aux topics, nous avons le programme `mosquitto_sub`. Pour publier des messages, nous avons le programme `mosquitto_pub`.

22. Ouvrez deux terminaux et tapez respectivement dans chacun les commandes suivantes :

```
mosquitto_sub -t unSujet
```

```
mosquitto_pub -t unSujet -m "unMessage"
```

23. puis donnez une interprétation du résultat obtenu ?

24. quel est le rôle de l'argument `-t` ?

25. quel est le rôle de l'argument `-m` ?

26. consultez le man de `mosquitto` ?

27. Le fichier de configuration par défaut de `mosquitto` est `mosquitto.conf`, localisez-le et donnez le chemin ?

28. Examiner le fichier et commenter sa structure ?

Serveurs MQTT publics

Les serveurs MQTT, `mqtt.eclipse.org`, `test.mosquitto.org`, `dev.rabbitmq.com`, ou encore `broker.hivemq.com` sont publics.

Pour exploiter ces sites, nous allons utiliser le client NodeJS `mqtt.js`

<https://github.com/mqttjs/MQTT.js>

29. Installez Nodejs sur votre système (par exemple sous Ubuntu, voir <https://doc.ubuntu-fr.org/nodejs>)

30. Ensuite lancez la commande `npm install -g mqtt`

31. Donnez la commande qui permet de s'abonner à un sujet avec un sujet et un nom de serveur public. Par exemple, abonnez-vous au sujet spécial `$/SYS` pour obtenir des informations techniques sur le serveur MQTT `mqtt.eclipseprojects.io` ?

32. attention : la sortie est longue, faite `Ctrl+C` et interprétez le résultat

```
mosquitto_sub -v -h mqtt.eclipse.org -t '$SYS/#'
```

33. Lancez la commande suivante et interprétez le résultat :

```
$ mqtt subscribe -v -h iot.eclipse.org '#' 2>/dev/null | head -10 | tee mqtt.log
```

34. Etant donné que nous sommes sur un serveur MQTT public, sans authentification, on constate qu'il est possible de prendre un sujet existant et son contenu et le rejouer à l'aide de la commande `mqtt publish`. Quelle sera la conséquence de cette commande ?

Réponse : Une usurpation du périphérique producteur et la falsification des messages envoyés, ce qui peut avoir de graves conséquences sur des systèmes embarqués connectés. Tous les clients consommateurs de ce sujet seront dupés puisque rien n'indique dans le flux que le producteur n'est plus le même).