

Matériel nécessaire :

- RPI 3 connectée à internet (distribution : "Raspbian Buster with desktop and recommended software" 2020-02-13)
- PC équipé de putty, wireshark et MQTT-FX.

Installation de Mosquitto sur Raspberry

Avant d'installer Mosquitto, mettre à jour votre système :

`sudo apt update && sudo apt upgrade -y`

Installer le broker Mosquitto :

`sudo apt-get install mosquitto`

Installer le client mosquitto pour effectuer des tests :

`sudo apt-get install mosquitto-clients`

Le client mosquitto permet de souscrire à des topics (**`mosquitto_sub`**), de publier des messages (**`mosquitto_pub`**) et de configurer les utilisateurs (**`mosquitto_passwd`**).



Tester le broker mosquitto

Ouvrir deux terminales.

Dans le premier, souscrire au topic « sensor/temperature » :

`mosquitto_sub -h localhost -t "g133/temperature"`

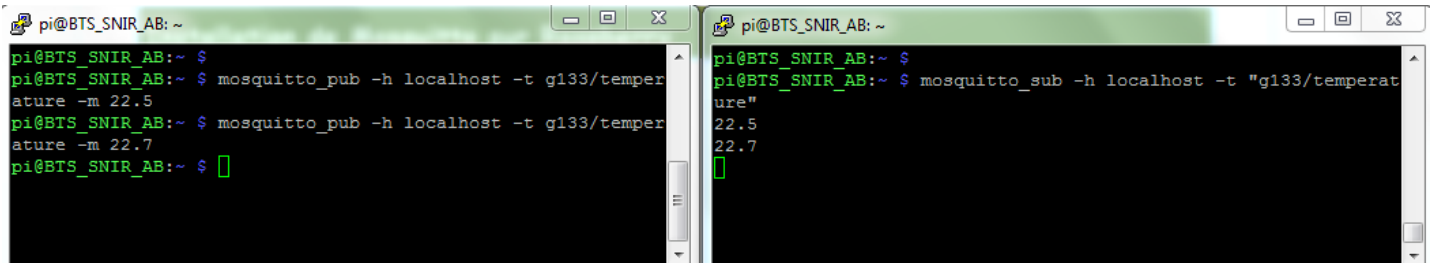
Se connecte au serveur (-h) localhost

Sélectionner le topic (-t) g133/temperature

Dans le deuxième terminale, publier une valeur (-m) 22.5 sur le même topic « sensor/temperature » :

`mosquitto_pub -h localhost -t g133/temperature -m 22.5`

Le résultat est le suivant :



Terminale correspondant au « Publisher »

Terminale correspondant au « Subscriber »

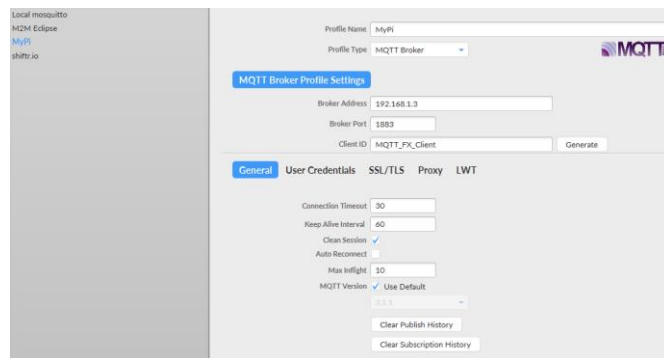
Tester la qualité de service (QoS) avec MQTT

Par défaut un message est envoyé avec un **QoS niveau 0** « At most once » (il sera délivré au plus une fois).

Nous allons analyser avec Wireshark les échanges entre le broker Mosquitto installé sur la raspberry et un publisher installé sur un PC (logiciel MQTT-FX). Mais pour lancer une capture, il faut d'abord déterminer un filtre à appliquer (protocole et port utilisé)

Configurer les échanges MQTT entre PC et raspberry

Lancer sur votre PC le logiciel MQTT-FX et créer un nouveau profil de connexion afin de cibler le broker mosquitto de votre raspberry.



Publier sur le topic « g133/temperature » après avoir lancé un client mosquitto sur votre raspberry qui souscrit au même topic.

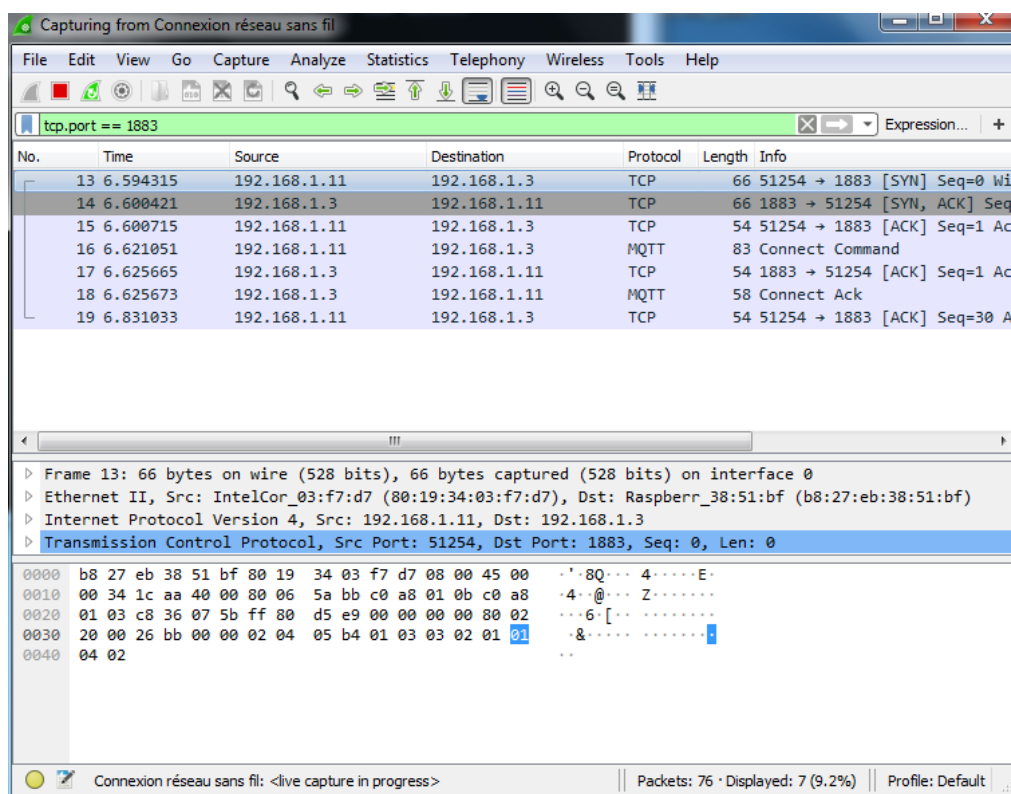
Vous devez obtenir le résultat suivant :

```
pi@BTS_SNIR_AB:~$ mosquitto_sub -h localhost -t "g133/temperature"
22.795265
```

Lancer la capture avec WireShark

Afin de filtrer les données sous WireShark, il faut déterminer le protocole et le port utiliser par le broker mosquitto. Pour cela, vous devez vous mettre en root (*sudo su*) et utiliser la commande *netstat -taupen*. Déconnecter MQTT-FX et lancer wireshark sur le pc (publisher) et ajouter un filtre : *tcp.port==1883*. Relancer le MQTT-FX et capturer les échanges lors de la connection au broker.

Vous devez obtenir le résultat suivant :



Refaire une nouvelle capture lorsque le publisher publie une température, vous devez obtenir le résultat suivant :

No.	Time	Source	Destination	Protocol	Length	Info
2	3.689355	192.168.1.11	192.168.1.3	MQTT	84	Publish Message [/g133/temperature]
4	3.742305	192.168.1.3	192.168.1.11	TCP	54	1883 → 51254 [ACK] Seq=1 Ack=31 Win=229 Len=0

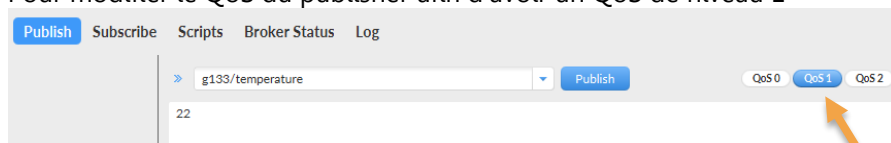
Msg Len: 28	
Topic Length: 17	
Topic: /g133/temperature	
Message: 23.042217	

0000	b8 27 eb 38 51 bf 80 19 34 03 f7 d7 08 00 45 00	..8Q... 4.....E..
0010	00 46 1d ab 40 00 80 06 59 a8 c0 a8 01 0b c0 a8	..F..@... Y.....
0020	01 03 c8 36 07 5b ff 80 d6 0d 6d 93 ac 6f 50 18	...6.[...m..oP..
0030	11 19 4d ef 00 00 30 1c 00 11 2f 67 31 33 33 2f	..M..0.../g133/
0040	74 65 6d 70 65 72 61 74 75 72 65 32 33 2e 30 34	temperat ure23.04
0050	32 32 31 37	2217

Par défaut, le QoS du logiciel MQTT-FX est de niveau 0. On constate qu'avec ce niveau de qualité de service, les informations sont simplement transmises et validées uniquement par TCP.

Modifier le QoS

Pour modifier le QoS du publisher afin d'avoir un QoS de niveau 1



Relancer une capture wireshark, vous devez obtenir le résultat suivant :

Capturing from Connexion réseau sans fil						
No.	Time	Source	Destination	Protocol	Length	Info
21	3.628185	192.168.1.11	192.168.1.3	MQTT	86	Publish Message (id=7) [/g133/temperature]
22	3.634191	192.168.1.3	192.168.1.11	MQTT	58	Publish Ack (id=7)
24	3.833152	192.168.1.11	192.168.1.3	TCP	54	54856 → 1883 [ACK] Seq=33 Ack=5 Win=4372 Len=0

On constate que :

- ⇒ Le broker génère cette fois un accusé de réception MQTT. En effet, avec un QoS de niveau 1, les messages sont acquittés par le receveur, si cet acquittement n'arrive pas le message est retransmis.

Modifier le QoS au niveau 2 et refaire la manipulation. Vous devez obtenir le résultat suivant :

No.	Time	Source	Destination	Protocol	Length	Info
28	4.065770	192.168.1.11	192.168.1.3	MQTT	86	Publish Message (id=1) [/g133/temperature]
29	4.071181	192.168.1.3	192.168.1.11	MQTT	58	Publish Received (id=1)
30	4.072503	192.168.1.11	192.168.1.3	MQTT	58	Publish Release (id=1)
31	4.074624	192.168.1.3	192.168.1.11	MQTT	58	Publish Complete (id=1)
34	4.275376	192.168.1.11	192.168.1.3	TCP	54	55630 → 1883 [ACK] Seq=37 Ack=9 Win=4377 Len=0

On constate que :

- ⇒ Le broker génère cette fois un accusé de réception et confirme que le message a bien été ensuite effacé. En effet, avec un QoS de niveau 2, l'acquittement est lui-même validé afin de garantir que le message n'arrive qu'une seule fois.

Sécuriser le protocole MQTT (au niveau des accès)

Par défaut, n'importe quel client peut se connecter au broker Mosquitto. Ce n'est pas très gênant s'il n'y a que des informations basiques qui sont publiées (mesure de température, d'humidité, ...), mais si vous voulez commander des appareils, il faut sécuriser un minimum le broker.

Placez-vous dans le dossier de configuration de Mosquitto : `cd /etc/mosquitto/`

```
pi@raspberrypi:/etc/mosquitto $ ls -l
total 16
drwxr-xr-x 2 root root 4096 mars  1 17:08 ca_certificates
drwxr-xr-x 2 root root 4096 mars  1 17:08 certs
drwxr-xr-x 2 root root 4096 mars  1 17:08 conf.d
-rw-r--r-- 1 root root 348 nov. 16 18:56 mosquitto.conf
```

Mosquitto crée un fichier qui contient le nom de l'utilisateur et son mot de passe (crypté). Pour ajouter un nouvel utilisateur exécutez la commande ***mosquitto_passwd -c nomfichier nomutilisateur***. Remplacez nomfichier par le nom de fichier qui contiendra les identifiants et nomutilisateur, le nom de l'utilisateur à ajouter : ***sudo mosquitto_passwd -c users snir_ab***

Mosquitto demande la saisie du mot de passe (2 fois).

Un nouveau fichier contenant les utilisateurs est créé :

```
pi@raspberrypi:/etc/mosquitto $ sudo mosquitto_passwd -c users snir_ab
Password:
Reenter password:
pi@raspberrypi:/etc/mosquitto $ ls -l
total 20
drwxr-xr-x 2 root root 4096 mars  1 17:08 ca_certificates
drwxr-xr-x 2 root root 4096 mars  1 17:08 certs
drwxr-xr-x 2 root root 4096 mars  1 17:08 conf.d
-rw-r--r-- 1 root root 348 nov. 16 18:56 mosquitto.conf
-rw-r--r-- 1 root root 117 mars  1 17:47 users
```

Ouvrez le fichier de configuration de Mosquitto avec nano (sudo obligatoire sur Linux) : ***sudo nano mosquitto.conf***

Ajoutez à la fin la commande allow_anonymous false et indiquez le chemin vers le fichier des mots de passe :

allow_anonymous false

password_file /etc/mosquitto/users

Maintenant il ne reste plus qu'à **redémarrer Mosquitto** pour prendre en compte les modifications : ***sudo service mosquitto restart***

Utilisez la commande Linux cat pour vérifier que le nouvel utilisateur a bien été ajouté au fichier de configuration :

```
pi@BTS_SNIR_AB:/etc/mosquitto $ cat users
snir_ab:$6$mGMpho1qC3wTkC3d$Dc3XBWXIuPl0MLSX2N05/X6EtzOd8ivz2U5aHqGLpm9R7vSXnNRV
/oyGm59iiiPCE63Mvs1dkCKc3pZ7g3a0A==
```

Maintenant, il faudra s'authentifier pour pouvoir publier ou souscrire à un Topic. En ligne de commande, on dispose de l'option -u pour spécifier l'utilisateur et -P pour le mot de passe. Ce qui donne pour publier un message :

mosquitto_pub -h localhost -t test -m demo -u nomutilisateur -P motdepasse

Et pour souscrire : ***mosquitto_sub -h localhost -t "test" -v -u nomutilisateur -P motdepasse***

Au niveau du logiciel MQTT-FX, il faut modifier les paramètres de connexions :

The screenshot shows the MQTT-FX web interface. On the left is a sidebar with links: 'Local mosquitto', 'M2M Eclipse', 'MyPi' (highlighted), and 'shiftr.io'. The main area is titled 'MQTT Broker Profile Settings' and contains fields for 'Profile Name' (MyPi), 'Profile Type' (MQTT Broker), 'Broker Address' (192.168.1.3), 'Broker Port' (1883), and 'Client ID' (MQTT_FX_Client). Below this is a tabbed interface with 'General', 'User Credentials', 'SSL/TLS', 'Proxy', and 'LWT'. The 'User Credentials' tab is active, showing 'User Name' (snir_ab) and 'Password' (masked with dots). An orange arrow points to the 'User Credentials' tab.