

### Q1.

當訂閱的 topic 為 /a/test 時，能不能接收到 topic 為 a/test 的訊息？為什麼？

(hint：比較兩個 topic 的層數)

不能。在 MQTT 協議中，topic 的命名是層級式的，使用斜線 '/' 分隔不同層級。當訂閱的 topic 為 /a/test 時，它將僅接收與該特定 topic 完全匹配的訊息，並不會接收與之類似但不完全相同的 topic (a/test) 的訊息。

### Q2.

Msg format : Place/Floor/Sensor Type/Time/ 共 36 種

- Place : house1、house2
- Floor : firstfloor、secondfloor、roof
- Sensor Type : temp、humid、brightness
- Time : day、night

若只想收到以下內容，那麼 subscriber 的 topic 分別要定為什麼？

1. 所有 roof 在 day 的 brightness
2. house1 firstfloor 的所有內容
3. house2 在 night 的所有內容
4. 所有 roof 的所有內容

1. `+/roof/brightness/day`
2. `house1/firstfloor/#`
3. `house2/+/+/night`
4. `+/roof/#`

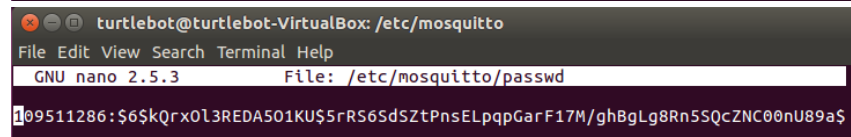
### Q3.

排除匿名使用者截圖(參考簡報範例)，帳號設為學號。

```
turtlebot@turtlebot-VirtualBox:/etc/mosquitto$ mosquitto_sub -v -t '#'
Connection Refused: not authorised.
Connection Refused: not authorised.

turtlebot@turtlebot-VirtualBox:~$ mosquitto_pub -m test -t topic -u 109511286 -P 109511286
turtlebot@turtlebot-VirtualBox:~$

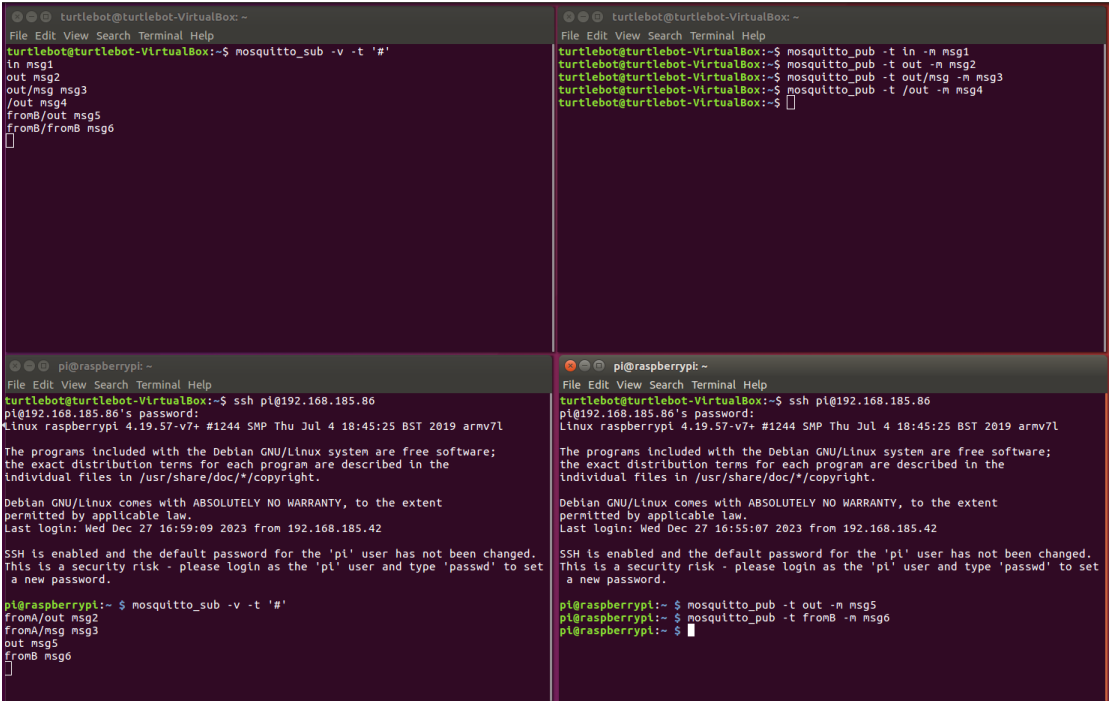
turtlebot@turtlebot-VirtualBox:/etc/mosquitto$ mosquitto_sub -v -t '#' -u 109511286 -P 109511286
topic test
```



```
turtlebot@turtlebot-VirtualBox: /etc/mosquitto
File Edit View Search Terminal Help
GNU nano 2.5.3 File: /etc/mosquitto/passwd
109511286:$6$KQrX0L3REDA501KU$5rRS6SdSZtPnsELpqGarF17M/ghBgLg8Rn55QcZNC00nU89a$
```

Q4.

Broker : topic 、 msg	Broker 1 (A) sub 顯示	Broker 2 (B) sub 顯示
A : in 、 msg1	in msg1	-
A : out 、 msg2	out msg2	fromA/out msg2
A : out/msg 、 msg3	out/msg msg3	fromA/msg msg3
A : /out 、 msg4	/out msg4	-
B : out 、 msg5	fromB/out msg5	out msg5
B : fromB 、 msg6	fromB/fromB msg6	fromB msg6



Q5.

在 MQTT bridge configuration 中，QoS level 有哪幾種?分別描述不同 level 的功能。

MQTT 定義了 0, 1 和 2 三個層級的品質設定。

0：最多傳送一次 (at most once) → 不保證訊息會送達。

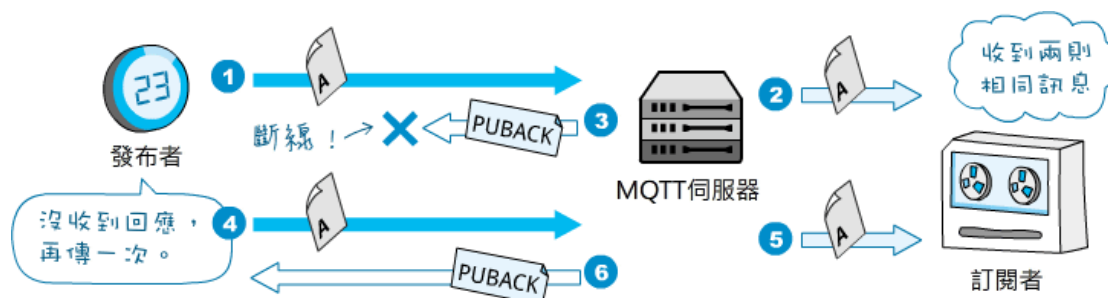
即便 MQTT 裝置之間的訊息傳遞基於 TCP/IP，但因為在網路階層架構中，

MQTT 程式屬於應用層，在將訊息交付給下一層(TCP)處理的過程中，MQTT 程式並不清楚最底層的網路是否壅塞甚至斷線，所以 MQTT 並不保證能交付訊息。

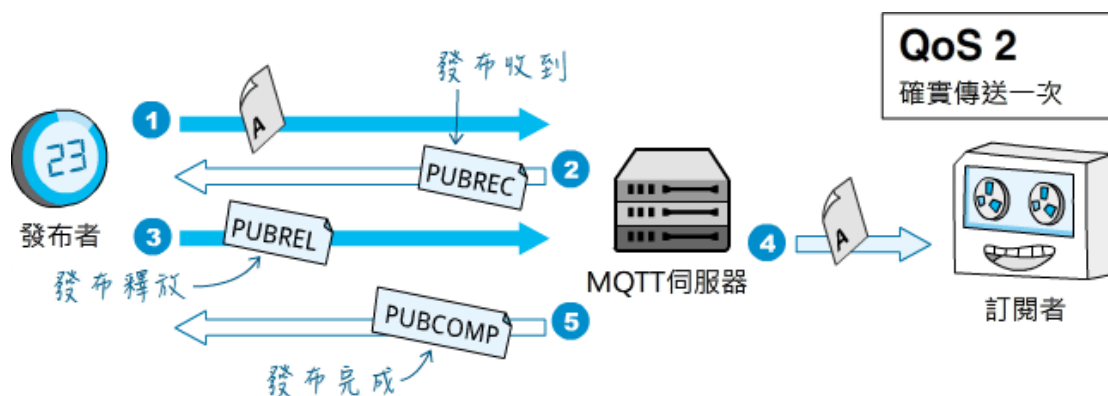


1：至少傳送一次 (at least once) → 收訊者收到訊息後，會回覆 PUBACK 訊號給發布者，確認有收到訊息。若一段時間過後發布者未收到 PUBACK 訊號，便會再重新傳送一次訊息。但如果未收到 PUBACK 訊號的原因是發布者因連線問題而漏接回應，發布者便會誤以為對方沒收到訊息而重新傳遞，導致收訊者重複收到相同訊息。

在「品質」設定為 1 或 2 的情況下，MQTT 訊息檔頭後面會夾帶一個稱為 packetId (封包 ID) 的識別碼，收訊者回應的 PUBACK 訊息也會傳回相同的識別碼提供發布者確認。



2：確實傳送一次 (exactly once) → 收訊者收到訊息後，會回覆 PUBREC 訊號給發布者，並且暫存訊息的封包識別碼，以避免重複收到相同訊息。當發布者收到 PUBREC 後，將傳送 PUBREL 給收訊者，收訊者收到後，會將訊息傳送給訂閱者，再回應 PUBCOMP 給發布者並刪除暫存的訊息。



圖片來源：<https://swf.com.tw/?p=1015>

## Q6.

在 `mosquitto.conf` 中，對於設定 `cleansession` 的說明如下：

`cleansession [true | false]`

Set the clean session option for this bridge. Setting to `false` (the default), means that all subscriptions on the remote broker are kept in case of the network connection dropping. If set to `true`, all subscriptions and messages on the remote broker will be cleaned up if the connection drops. Note that setting to `true` may cause a large amount of retained messages to be sent each time the bridge reconnects.

If you are using bridges with `cleansession` set to `false` (the default), then you may get unexpected behavior from incoming topics if you change what topics you are subscribing to. This is because the remote broker keeps the subscription for the old topic. If you have this problem, connect your bridge with `cleansession` set to `true`, then reconnect with `cleansession` set to `false` as normal.

請理解 `cleansession` 參數的目的並回答以下問題：

1. 描述 `true` | `false` 分別的作用。
2. 如果 `cleansession` 設置為 `false`，而且你更改訂閱的主題，會出現什麼意外的行為？為什麼會發生這種情況？該怎麼解決？
3. 什麼是"保留消息 (retained messages)"，為什麼在 `cleansession` 設置為 `true` 時可能會出現大量的保留消息被發送的情況？

1. `cleansession` 參數用於設置橋接器 (bridge) 的 "clean session" 選項。設置為 `true` 表示在連接斷開後，所有訂閱和訊息都將在遠端代理 (remote broker) 被清除。設置為 `false` (默認值) 表示即使連接斷開，遠端代理仍會保留所有訂閱，以便在重新連接時繼續服務。
2. 當 `cleansession` 設置為 `false`，如果更改了訂閱的主題，可能會導致意外行為。這是因為遠端代理保留了舊主題的訂閱資訊，如果切換到新的主題，可能會收到舊主題的訊息，或者即使取消訂閱了舊主題，但仍然會收到相關訊息。解決這個問題的方法是先將 `cleansession` 設置為 `true` 連接一次，然後再設置為 `false` 正常連接。
3. "保留消息 (retained messages)" 是在 MQTT 中的一種機制，它允許發布者標記一條訊息為保留。當一條消息被標記為保留時，新訂閱者訂閱相應主題後，將立即收到最新的保留消息。當 `cleansession` 設置為 `true` 時，重新連接時可能會出現大量保留消息被發送的情況，因為在每次重新連接時，舊的保留消息將被重新發送給新的客戶端，這可能導致大量的消息流量。

## Q7. 心得:

這次實驗中曾經遇到 service mosquito status 一直出現 failed 的狀態，還以為實驗要做不出來了，結果把樹梅派重新 sudo reboot 個幾次之後問題就解決了，樹梅派真是個神奇的東西…

由於 MQTT 是為了物聯網而設計的協定，所以它所需頻寬較少，需要的硬體資源也是低的，很適合用在低功耗和網路帶寬有限的 IoT 環境，例如智慧家電。