

- MQTT là gì
  - Publish, subscribe
  - QoS
  - Retain
  - LWT
  - Reference:

## MQTT là gì

**MQTT** (Message Queuing Telemetry Transport) là một giao thức gửi dạng publish/subscribe sử dụng cho các thiết bị **Internet of Things** với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.

Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng **M2M**

MQTT cũng là giao thức sử dụng trong **Facebook Messenger**

Và MQTT là gì? Để có một cái nhìn toàn diện hoặc định nghĩa chi tiết, chỉ cần google “what is mqtt”, “mqtt slides” ... Trong bài viết này chúng ta chỉ nói ngắn gọn thôi, đủ để hiểu giao thức MQTT, bao gồm các định nghĩa “**subscribe**”, “**publish**”, “**qos**”, “**retain**”, “**last will and testament (lwt)**” - Và chỉ dành cho những ai đang muốn tìm hiểu về MQTT, không thì đọc toàn chữ thôi, mỗi mất lăm.

### **Publish, subscribe**

Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là mqtt client - gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như “/client1/channel1”, “/client1/channel2”. Quá trình đăng ký này gọi là “**subscribe**”, giống như chúng ta đăng ký nhận tin trên một kênh Youtube vậy. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu và kênh đã đăng ký. Khi một client gửi dữ liệu tới kênh đó, gọi là “**publish**”.

### **QoS**

Ở đây có 3 tùy chọn **QoS (Qualities of service)** khi “publish” và “subscribe”:

- **QoS0** Broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP, giống kiểu đem con bỏ chợ.
- **QoS1** Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
- **QoS2** Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.

**Xem thêm QoS:** <https://code.google.com/p/mqtt4erl/wiki/QualityOfServiceUseCases>

Một gói tin có thể được gửi ở bất kỳ QoS nào, và các client cũng có thể subscribe với bất kỳ yêu cầu QoS nào. Có nghĩa là client sẽ lựa chọn QoS tối đa mà nó có để nhận tin. Ví dụ, nếu 1 gói dữ liệu được publish với QoS2, và client subscribe với QoS0, thì gói dữ liệu được nhận về client này sẽ được broker gửi với QoS0, và 1 client khác đăng ký cùng kênh này với QoS 2, thì nó sẽ được Broker gửi dữ liệu với QoS2.

Một ví dụ khác, nếu 1 client subscribe với QoS2 và gói dữ liệu gửi vào kênh đó publish với QoS0 thì client đó sẽ được Broker gửi dữ liệu với QoS0. QoS càng cao thì càng đáng tin cậy, đồng thời độ trễ và băng thông đòi hỏi cũng cao hơn.

## Retain

Nếu RETAIN được set bằng 1, khi gói tin được publish từ Client, Broker **PHẢI** lưu trữ lại gói tin với QoS, và nó sẽ được gửi đến bất kỳ Client nào subscribe cùng kênh trong tương lai. Khi một Client kết nối tới Broker và subscribe, nó sẽ nhận được gói tin cuối cùng có RETAIN = 1 với bất kỳ topic nào mà nó đăng ký trùng. Tuy nhiên, nếu Broker nhận được gói tin mà có QoS = 0 và RETAIN = 1, nó sẽ hủy tất cả các gói tin có RETAIN = 1 trước đó. Và phải lưu gói tin này lại, nhưng hoàn toàn có thể hủy bất kỳ lúc nào.

Khi publish một gói dữ liệu đến Client, Broker phải se RETAIN = 1 nếu gói được gửi như là kết quả của việc subscribe mới của Client (giống như tin nhắn ACK báo subscribe thành công). RETAIN phải bằng 0 nếu không quan tâm tới kết quả của việc subscribe.

## LWT

Gói tin LWT (last will and testament) không thực sự biết được Client có trực tuyến hay không, cái này do gói tin KeepAlive đảm nhận. Tuy nhiên gói tin LWT như là thông tin điều gì sẽ xảy đến sau khi thiết bị ngoại tuyến.

## Một ví dụ

Tôi có 1 cảm biến, nó gửi những dữ liệu quan trọng và rất không thường xuyên. Nó có đăng ký trước với Broker một tin nhắn lwt ở topic **/node/gone-offline** với tin nhắn **id** của nó. Và tôi cũng đăng ký theo dõi topic **/node/gone-offline**, sẽ gửi SMS tới điện thoại tôi mỗi khi nhận được tin

nhấn nào ở kênh mà tôi theo dõi. Trong quá trình hoạt động, cảm biến luôn giữ kết nối với Broker bởi việc luôn gửi gói tin keepAlive. Nhưng nếu vì lý do gì đó, cảm biến này chuyển sang ngoại tuyến, kết nối tới Broker timeout do Broker không còn nhận được gói keepAlive. Lúc này, do cảm biến của tôi đã đăng ký LWT, do vậy broker sẽ đóng kết nối của Cảm biến, đồng thời sẽ publish một gói tin là Id của cảm biến vào kênh **/node/gone-offline**, dĩ nhiên là tôi cũng sẽ nhận được tin nhắn báo cái Cảm biến yêu quý của mình đã ngoại tuyến.

## Ngắn gọn

Ngoài việc đóng kết nối của Client đã ngoại tuyến, gói tin LWT có thể được định nghĩa trước và được gửi bởi Broker tới kênh nào đó khi thiết bị đăng ký LWT ngoại tuyến.

## Reference:

- <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
- <http://bb-smartsensing.com/basics-of-mqtt/>
- <http://stackoverflow.com/questions/17270863/>
- <http://tuanpmt.github.io/what-is-mqtt/>